

Quantum GIS

Manuel
Installation, Utilisation, Programmation

Version 1.0.0 *'Kore'*

Préambule

Ce document est le manuel officiel pour l'installation, l'utilisation et la programmation du logiciel Quantum GIS. Les logiciels et le matériel décrits dans ce document sont pour la plupart des marques déposées et donc soumises à des obligations légales. Quantum GIS est distribué sous la Licence publique générale GNU (GPL). Vous trouverez plus d'informations sur la page internet de Quantum GIS http://qgis.osgeo.org.

Les détails, données, résultats, etc. inclus dans ce document ont été écrits et vérifiés au mieux des connaissances des auteurs et des éditeurs. Néanmoins, des erreurs dans le contenu sont possibles.

Ainsi l'ensemble des données ne sauraient faire l'objet d'une garantie. Les auteurs et les éditeurs ne sauraient être responsables de tout dommage direct, indirect, secondaire ou accessoire découlant de l'utilisation de ce manuel. Les éventuelles corrections sont toujours les bienvenues.

Ce document a été rédigé avec LaTeX. Les sources sont disponibles en code LaTeXvia http://wiki.qgis.org/qgiswiki/DocumentationWritersCornersubversion et en PDF via http://qgis.osgeo.org/documentation/manuals.html. Des versions traduites peuvent être téléchargées via la section de documentation du projet QGIS. Pour plus d'informations sur les manières de contribuer à ce document et à sa traduction, veuillez visiter http://wiki.qgis.org/qgiswiki/DocumentationWritersCorner

Références de ce document

Ce document contiens des références internes et externes sous forme de lien. Cliquer sur un lien interne provoque un déplacement dans le document, tandis que cliquer sur un lien externe ouvrira une adresse internet dans le navigateur par défaut. En PDF, les liens internes seront indiqués en bleu et les externes en rouge. En HTML, le navigateur affiche et gère les deux types de liens de la même façon.

Auteurs et éditeurs :

Tara Athan
Radim Blazek
Godofredo Contreras
Otto Dassau
Martin Dobias
Jürgen E. Fischer
Stephan Holl
Marco Hugentobler
Magnus Homann
Lars Luthman
Gavin Macaulay
Werner Macho
Tyler Mitchell
Brendan Morely
Gary E. Sherman
Tim Sutton
David Willis

Traducteurs :

Benjamin Bohard
Jeremy Garniaux
Yves Jacolin
Stéphane Morel
Jean Roc Morreale
Marie Silvestre
Cyril de Ruz

Contributeurs à la version française :

Ludovic Granjon (aide à la compilation)

Remerciement à Tisham Dhar pour avoir préparé l'environnement de documentation initial pour MS Windows, à Tom Elwertowski et William Kyngesburye pour la section d'installation sur Mac OS X et à Carlos Carlos Dávila, Paolo Cavallini et Christian Gunning pour les révisions. Si nous avons négligé de citer ici le nom d'un contributeur, veuillez accepter nos excuses pour cet oubli.

Internet : http://qgis.osgeo.org

Table des matières

Table des figures

Liste des tableaux

Astuce QGIS

1 Avant-propos

Bienvenue dans le monde merveilleux des Systèmes d'Information géographiques (SIG) ! Quantum GIS est un SIG libre qui a débuté en mai 2002 et s'est établi en tant que projet en juin 2002 sur SourceForge. Nous avons travaillé dur pour faire de ce logiciel SIG (qui sont traditionnellement des logiciels propriétaires assez coûteux) un choix viable pour toute personne ayant un ordinateur. QGIS est utilisable sur la majorité des Unix, Mac OS X et Windows. QGIS utilise la bibliothèque logicielle Qt 4 (`http://www.trolltech.com`) et le langage C++, ce qui ce traduit par une interface graphique simple et réactive.

QGIS se veut simple à utiliser, fournissant des fonctionnalités courantes. Le but initial était de fournir un visualisateur de données SIG, QGIS a depuis atteint un stade dans son évolution où beaucoup y recourent pour leurs besoins journaliers. QGIS supporte un grand nombre de formats raster et vecteur, avec un support de nouveaux formats facilités par l'architecture des modules d'extension (lisez l'Annexe `appdx_data_formats` pour une liste complète des formats actuellement supportés)

QGIS est distribué sous la licence GPL. Ceci vous permet de pouvoir regarder et modifier le code source, tout en vous garantissant un accès à un programme SIG sans coût et librement modifiable. Vous devez avoir reçu une copie complète de la licence avec votre exemplaire de QGIS, vous la trouverez également dans l'Annexe N.

Astuce 1 DOCUMENTATION À JOUR
La dernière version de ce document est disponible sur `http://download.osgeo.org/qgis/doc/manual/`, ou dans la section documentation du site de QGIS `http://qgis.osgeo.org/documentation/`

1.1 Fonctionnalités

QGIS offre beaucoup d'outils SIG standards par défaut et via les extensions. Voici un bref résumé en six catégories qui vous donnera un premier aperçu.

Visualiser des données

Vous pouvez afficher et superposer des couches de données rasters et vecteurs dans différents formats et projections sans avoir à faire de conversion dans un format commun. Les formats supportés incluent :

– les tables spatiales de PostgreSQL/PostGIS, les formats vecteurs supportés par la bibliothèque OGR installée [1], ce qui inclue les fichiers de forme ESRI (shapefiles), MapInfo, STDS et GML.
– les formats raster supportés par la bibliothèque GDAL (Geospatial Data Abstraction Library) tel que GeoTiff, Erads Img., ArcInfo Ascii Grid, JPEG, PNG...
– les formats raster et vecteur provenant des bases données GRASS.

[1] les formats de base de données Oracle et MySQL sont supportés par OGR mais pas encore par QGIS.

– les données spatiales provenant des services réseaux compatibles OGC comme le Web Map Service (WMS) ou le Web Feature Service (WFS).

Parcourir les données et créer des cartes

Vous pouvez créer des cartes et les parcourir de manière interactive avec une interface abordable. Les outils disponibles dans l'interface sont :

– projection à la volée
– créateur de carte
– panneau de navigation
– marque-pages spatiaux
– identifier et sélectionner des entités
– voir, éditer et rechercher des attributs
– étiquetage des entités
– changer la symbologie des rasters et vecteurs
– ajouter une couche de graticule
– ajout d'une barre d'échelle, d'une flèche indiquant le nord et d'une étiquette de droits d'auteur
– sauvegarde et chargement de projets

Créer, éditer, gérer et exporter des données

Vous pouvez créer, éditer, gérer et exporter des données vecteurs dans plusieurs formats. Les données raster doivent être importées dans GRASS pour pouvoir être éditées et exporter dans d'autres formats. QGIS permet ce qui suit :

– outils de numérisation pour les formats d'OGR et les couches vecteurs de GRASS
– créer et éditer des fichiers de forme (shapefiles) et les couches vecteurs de GRASS
– géocodifier des images avec l'extension de géoréférencement
– outils d'import/export du format GPX pour les données GPS, avec la conversion des autres formats GPS vers le GPX ou l'envoi/réception directement vers une unité GPS
– créer des couches PostGIS à partir de fichiers de forme (shapefiles) avec l'extension SPIT
– gérer les attributs de tables des couches vecteurs grâce à l'extension de gestion des tables

Analyser les données

Vous pouvez opérer des analyses spatiales sur des données PostgreSQL/PostGIS et autres formats OGR en utilisant l'extension ftools. QGIS permet actuellement l'analyse vectorielle, l'échantillonnage, la gestion de la géométrie et des bases de données. Vous pouvez aussi utiliser les outils GRASS intégrés qui comportent plus de 300 modules (voir la section 9)

Publier une carte sur Internet

QGIS peut être employé pour exporter des données vers un mapfile et le publiqer sur internet via un serveur web employant l'UMN MapServer. QGIS peut aussi servir de client WMS/WFS ou comme un serveur WMS.

Etendre les fonctionnalités de QGIS grâçe à des extensions

QGIS peut être adaptée à vos besoins particuliers du fait de son architecture d'extensions. QGIS fournit des bibliothèques qui peuvent être employées pour créer des extensions, vous pouvez même créer de nouvelles applications en C++ ou python !

– **Extensions principales**
  Ajouter une couche WFS
  Ajouter une couche de texte délimité
  Capture de coordonnées
  Décorations (Étiquette de droit d'auteur, flèche indiquant le nord et barre d'échelle)
  Georérérencement
  Convertisser Dxf2Shp
  Outils GPS
  Intégration de GRASS
  Créateur de graticules
  Exenstion d'interpolation
  Convertisseur de couche OGR
  impression rapide
  SPIT, outil d'importation de Shapefile vers PostgreSQL/PostGIS
  Exportation vers Mapserver
  Terminal Python
  Installateur d'extension Python

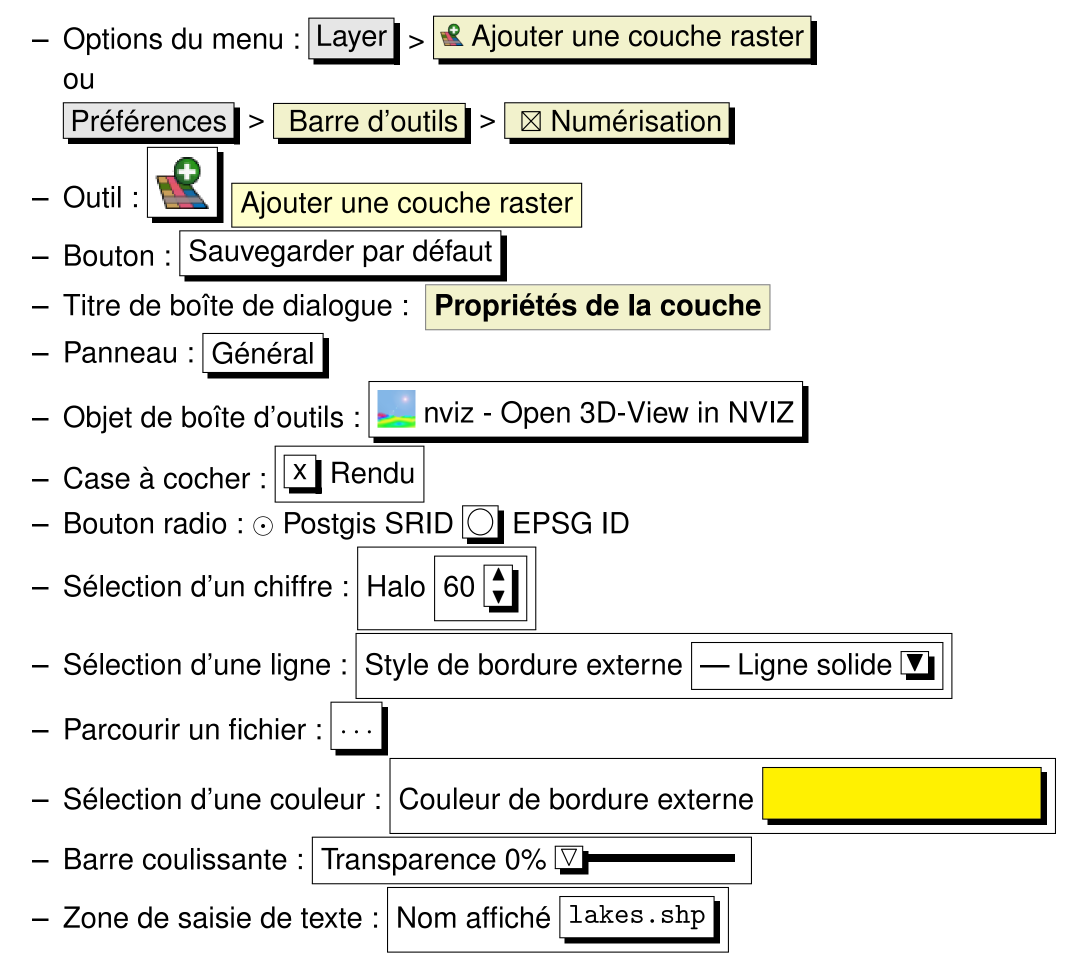
– **Extensions Python**
  QGIS offre un nombre croissant d'extensions complémentaires en python fourni par la communauté. Ces extensions sont entreposées dan le répertoire PyQGIS et peuvent être facilement installé en utilisant l'extension d'installation python (Voir Section 11).

1.2 Conventions

Cette section décrit les symboles qui ponctuent ce manuel, les conventions graphiques sont les suivantes

Conventions pour l'interface

Les styles de conventions de l'interface (GUI) dans le texte ressemblent autant que possible à l'apparence du logiciel, l'objectif étant de permettre à l'utilisateur de repérer plus facilement les éléments mentionnés dans les instructions.



- Options du menu : Layer > Ajouter une couche raster
  ou
  Préférences > Barre d'outils > ⊠ Numérisation
- Outil : Ajouter une couche raster
- Bouton : Sauvegarder par défaut
- Titre de boîte de dialogue : **Propriétés de la couche**
- Panneau : Général
- Objet de boîte d'outils : nviz - Open 3D-View in NVIZ
- Case à cocher : X Rendu
- Bouton radio : ⊙ Postgis SRID ◯ EPSG ID
- Sélection d'un chiffre : Halo 60
- Sélection d'une ligne : Style de bordure externe — Ligne solide
- Parcourir un fichier : ...
- Sélection d'une couleur : Couleur de bordure externe
- Barre coulissante : Transparence 0%
- Zone de saisie de texte : Nom affiché `lakes.shp`

Une ombre indique un élément de l'interface qui peut être cliqué.

Conventions de texte ou de clavier

Le manuel se réfère aussi à des conventions pour le texte, les commandes du clavier et l'encodage pour définir les entités, les classes et les méthodes. Elles ne correspondent pas à l'apparence réelle.

- Hyperliens : `http://qgis.org`

- Simple touche : appuyez sur p
- Combinaisons de touches : appuyez sur Ctrl+B, signifie qu'il faut rester en appui sur la touche Contrôle (Ctrl) tout en pressant la touche B.
- Nom d'un fichier : `lakes.shp`
- Nom d'une classe : **NewLayer**
- Méthode : *classFactory*
- Serveur : *myhost.de*
- Texte pour l'utilisateur : `qgis --help`

Les codifications sont indiquées par une police à taille fixe :

```
PROJCS["NAD_1927_Albers",
  GEOGCS["GCS_North_American_1927",
```

Instructions spécifiques à une plateforme

GUI sequences and small amounts of text can be formatted inline : Clic { Fichier QGIS} > Quitter pour fermer QGIS.

Cela indique que sous Windows, Linux et les plateformes Unix il faudra d'abord cliquer sur Fichier puis dans la liste déroulane sur Quitter, alors que sous Mac il faudra cliquer sur le menu Qgis. De grandes portions de textes peuvent être présentées en liste :

- faites ceci ;
- faites cela ;
- faites autre chose.

ou comme des paragraphes :

Faites ceci et cela. Puis cela et ceci, ensuite ceci et cela pour obtenire ceci et cela, etc.

Faites ceci et cela. Puis cela et ceci, ensuite ceci et cela pour obtenire ceci et cela, etc.

Les aperçus d'écrans ont été pris sous différentes plateformes, un icône à la fin de la légende de la figure indique le système en question.

2 Introduction au SIG

Un Système d'Information Géographique (SIG)**?**[2] est une collection de logiciels qui vous permettent de créer, visualiser, rechercher et analyser des données géospatiales. Ces données se réfèrent à des informations concernant l'emplacement géographique d'une entité. Ceci implique souvent l'utilisation de coordonnées géographiques, tel qu'une valeur de latitude ou de longitude. Le terme donnée spatiale est également employé couramment, ainsi que : donnée géographique, donnée SIG, donnée cartographique, donnée de localisation, donnée de géométrie spatiale...

Les applications utilisant des données géospatiales réalisent une grande variété de fonctions. La création de carte est celle-là plus admise, les logiciels cartographiques prennent les données géospatiales et les restituent sous une forme visuelle, sur un écran d'ordinateur ou sur une page imprimée. Ces applications peuvent présenter des cartes statiques (une seule image) ou des cartes dynamiques qui peuvent être personnalisées par la personne regardant la carte via un logiciel bureautique ou une page internet.

Beaucoup de gens présument à tort que les applications géospatiales se limitent à la production de cartes alors que l'analyse des données est une autre importante fonction de ces logiciels. Quelques exemples d'analyses incluent les calculs :

1. de la distance entre deux points géographiques
2. de l'aire (p. ex., mètres carrés) d'une zone géographique
3. pour déterminer quelles entités se superposent sur d'autres entités
4. le taux de superposition entre entités
5. le nombre de points se situant à une certaine distance d'un autre
6. et beaucoup d'autres...

Cela semble peut-être simpliste, mais ils peuvent être appliqués à de nombreuses disciplines. Le résultat de ces analyses peut être affiché sur une carte mais plus généralement sous une forme tabulaire dans des rapports pour appuyer des décisions.

Le phénomène récent de services basés sur la localisation va introduire toutes sortes de nouvelles fonctionnalités, mais beaucoup seront issues de la conjugaison de cartes et d'analyses. Par exemple, si vous avez un téléphone portable qui affiche votre position. Si vous avez le bon type de logiciel, votre téléphone pourra vous signaler les restaurants se trouvant à une courte distance de marche. Bien que ce soit une nouvelle application des technologies géospatiales, il s'agit pour l'essentiel d'analyser des données géospatiales et de vous livrer les résultats.

[2]Ce chapitre est de Tyler Mitchell (`http://www.oreillynet.com/pub/wlg/7053`) et est utilisé sous une licence Creative Commons. Tyler est l'auteur de *Web Mapping Illustrated*, publié par O'Reilly, 2005.

2.1 Pourquoi tout cela est-il si récent ?

Et bien ça ne l'est pas. Il y a beaucoup de nouveaux appareils qui autorisent l'utilisation mobile de services géospatiaux. Beaucoup d'applications open source sont aussi disponibles, mais l'existence de matériels et logiciels dédiés à la géospatialisation n'est pas quelque chose de nouveau. Les récepteurs GPS (Global Positioning System) sont devenus courants mais sont utilisés dans certaines industries depuis plus d'une décennie. De la même manière, la cartographie bureautique et les outils d'analyse ont depuis longtemps représenté un important secteur commercial, consacré à l'origine à des secteurs comme la gestion de ressources naturelles.

Ce qui est nouveau est la façon dont les appareils et applications sont utilisés et par qui. Les utilisateurs traditionnels étaient des géomaticiens hautement qualifiés ou des techniciens habitués à travailler avec des outils de CAO. Aujourd'hui les capacités de calculs des ordinateurs domestiques et des logiciels open source ont permis à une foule de passionnés, de professionnels, de développeurs internet, etc. d'interagir avec des données géospatiales. La courbe d'apprentissage a diminué, les coûts ont diminué tandis que la diffusion des technologies spatiales a augmenté.

Comment sont stockées ces informations ? Pour faire simple, il existe deux sortes de données géospatiales dont l'utilisation est très répandue de nos jours, ce à quoi s'ajoutent les données tabulaires qui continuent à être utilisées couramment par les applications géospatiales.

2.1.1 Les Données Raster

L'un des types de données géospatiales est qualifié de donnée raster/matricielle, ou plus communément un "raster". Les formes les plus facilement reconnaissables de donnée raster sont les images satellites numériques ou les photos aériennes. Les ombrages de pentes ou les modèles numériques de terrain sont également représentés en raster. Tout type de données cartographiques peut être représenté comme une donnée raster, mais il y a des limitations.

Un raster est une grille régulière qui se compose de cellules ou, dans le cas de l'imagerie, de pixels. Il y a un nombre déterminé de lignes et de colonnes. Chaque cellule a une valeur numérique et une certaine taille géographique (par exemple 30 x 30 mètres de surface).

De multiples rasters sont superposés pour afficher des images qui utilisent plus d'une valeur de couleur (c.-à-d. un raster pour chaque bande de valeurs de rouge, vert et bleu sont combinés pour créer une image couleur). L'imagerie satellite représente les données avec plusieurs bandes. Chacune de ces bandes est un raster distinct qui se superpose spatialement aux autres rasters, une bande détient des valeurs correspondant à certaines longueurs d'onde de la lumière. Comme vous pouvez l'imaginer, un gros raster prend plus d'espace-disque. Un raster avec de plus petites cellules fournira plus de détails, mais prendra plus de place. L'astuce est de trouver le juste équilibre entre la taille des cellules pour le stockage et la taille des cellules pour l'analyse ou la cartographie.

2.1.2 Les données vectorielles

Les données vectorielles sont également utilisées dans les applications géospatiales. Si vous êtes resté éveillé durant vos cours de trigonométrie et de géométrie, vous serez déjà familier avec quelques-unes des particularités des données vectorielles. Les vecteurs sont une façon de décrire un emplacement en utilisant une série de coordonnées, chaque coordonnée se référant à une localisation géographique utilisant un système de valeurs en x et en y.

On peut faire la comparaison avec un plan cartésien - vous savez, le diagramme de l'école qui présentait des axes x et y. Vous en avez sans doute eu recours pour des graphiques montrant la chute de votre épargne-retraite ou l'augmentation de votre taxe d'habitation, le concept est ici similaire et essentiel pour l'analyse et la représentation géospatiale.

Il y a différentes manières de représenter ces coordonnées qui dépendent de votre objectif, c'est un tout autre chapitre à étudier : celui des projections cartographiques. Les données vectorielles prennent trois formes, chacune progressivement plus complexe et s'appuyant sur la précédente.

1. les Points - une simple coordonnée (x y) qui représente un emplacement géographique ponctuel
2. les Lignes - plusieurs coordonnées (x1 y1, x2 y2, x3 y4, ... xn yn) reliées ensemble selon un ordre précis, tel que pour dessiner une ligne du point (x1 y1) au point (x2 y2) et ainsi de suite. Les parties qui se situent entre les points sont considérées comme des segments de ligne. Ils ont une longueur et la ligne peut avoir une direction suivant l'ordre des points. Techniquement, une ligne est une simple paire de points reliés ensemble tandis qu'une ficelle de ligne se compose multiples lignes qui sont connectées.
3. les Polygones - quand les lignes sont reliées par plus de deux points, avec le dernier point situé au même endroit que le premier, nous appelons le résultat un polygone. Un triangle, un cercle, un rectangle, etc. sont tous des polygones. La propriété clé des polygones est qu'ils ont une surface interne fixe.

3 Premiers Pas

Ce chapitre donne un aperçu rapide de l'installation de QGIS, de quelques échantillons de données provenant du site internet et du lancement d'une première session d'affichage de couches raster et vecteur.

3.1 Installation

L'installation de QGIS est très simple, des installateurs sont disponibles pour Windows et Mac OS X. Beaucoup de distributions Linux mettent à disposition des fichiers binaires (.rpm ou .deb) via leurs interfaces de gestion de logiciels. Obtenez les dernières informations concernant les paquets binaires sur le site de QGIS sur http://qgis.osgeo.org/download/.

Si vous avez besoin de compiler QGIS depuis les sources, le processus est documenté dans l'Annexe D pour MS Windows, Annexe E pour Mac OSX, Annexe F pour GNU/Linux. Les instructions d'installations sont distribuées avec le code source, mais aussi sur http://qgis.osgeo.org.

3.2 Échantillons de données

Le guide de l'utilisateur comporte une série d'exemples basée sur un échantillon de données inclu avec QGIS.

L'installateur Windows possède une option pour télécharger automatiquement l'échantillon de données. Si vous le cochez, les données seront téléchargées dans votre répertoire Mes Documents et placées dans un dossier GIS Database. Vous pouvez utiliser l'explorateur de fichiers Windows pour déplacer ce dossier à votre convenance. Si vous n'avez pas coché cette option durant l'installation, vous avez plusieurs solutions :

- utiliser des données que vous avez déjà ;
- télécharger l'échantillon sur le site de QGIS http://qgis.osgeo.org/download ; ou
- désinstaller QGIS puis réinstaller en cochant la case de téléchargement .

Pour GNU/Linuxet Mac OSX il n'y a pas encore de paquet disponible sous forme de rpm, deb ou dmg. Pour utiliser l'échantillon de données, téléchargez le fichier qgis_sample_data en ZIP ou archive TAR depuis http://download.osgeo.org/qgis/data/ et décompressez-le ou désarchivez-le dans votre système. Le jeu de données sur l'Alaska comporte toutes les données SIG qui ont servi à la préparation des captures d'écran et des exemples qui figurent dans cet ouvrage. La projection est l'Alaska Albers Equal Area avec pour unité le pied dont le code EPSG est le 2964.

```
PROJCS["Albers Equal Area",
GEOGCS["NAD27",
DATUM["North_American_Datum_1927",
```

```
SPHEROID["Clarke 1866",6378206.4,294.978698213898,
AUTHORITY["EPSG","7008"]], TOWGS84[-3,142,183,0,0,0,0], AUTHORITY["EPSG","6267"]],
PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],
UNIT["degree",0.0174532925199433,
AUTHORITY["EPSG","9108"]], AUTHORITY["EPSG","4267"]],
PROJECTION["Albers_Conic_Equal_Area"],
PARAMETER["standard_parallel_1",55],
PARAMETER["standard_parallel_2",65],
PARAMETER["latitude_of_center",50],
PARAMETER["longitude_of_center",-154],
PARAMETER["false_easting",0],
PARAMETER["false_northing",0],
UNIT["us_survey_feet",0.3048006096012192]]
```

Si vous désirez utiliser QGIS comme une interface à GRASS, vous trouverez une sélection d'échantillons de localisations (e.g. Spearfish ou South Dakota) sur le site officiel de GRASS `http://grass.osgeo.org/download/data.php`.

3.3 Session d'essai

Maintenant que vous avez QGIS d'installé et un échantillon de données disponible, nous voudrions vous faire une courte démonstration. Vous allez visualiser une couche raster et une couche vecteur. Nous allons utiliser la couche raster landcover `qgis_sample_data/raster/landcover.img` et la couche vectorielle représentant les lacs `qgis_sample_data/gml/lakes.gml`.

Démarrer QGIS

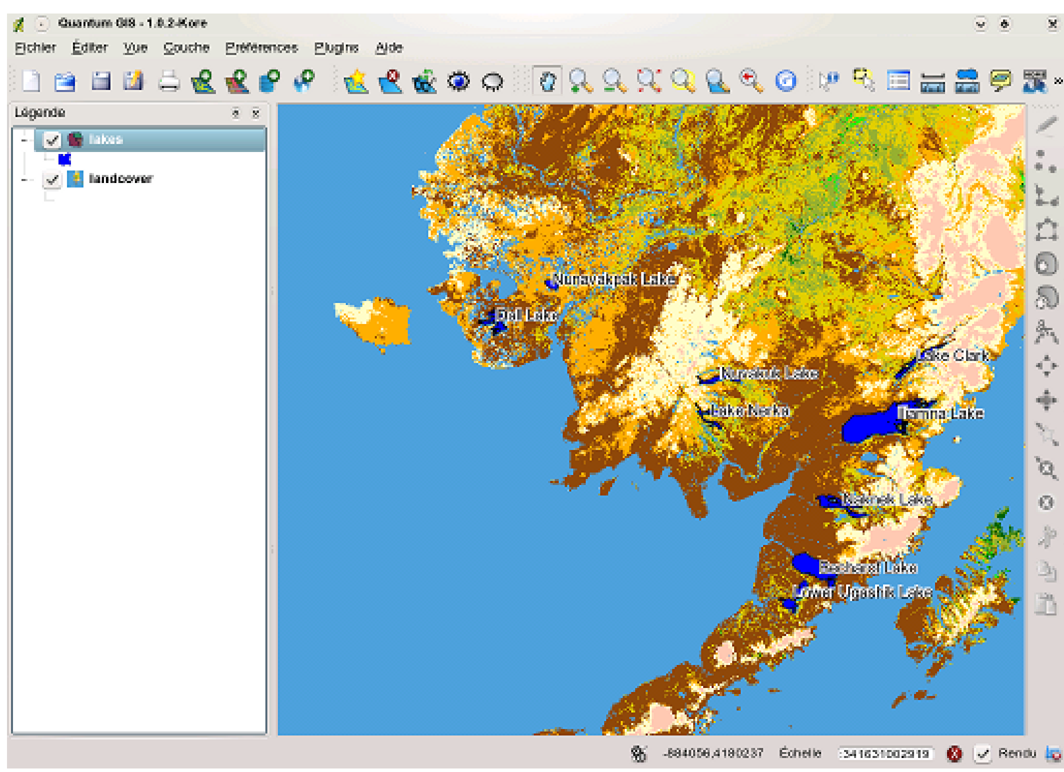
– Démarrer QGIS en tapant : `qgis` dans une ligne de commande console.
– Démarrez QGIS en utilisant le menu Démarrer ou un raccourci sur le bureau, ou double-cliquez sur un fichier de projet QGIS.
– Double-cliquez sur l'icône de QGIS dans votre répertoire Applications.

Charger les couches raster et vecteur depuis le jeu de données

1. Cliquez sur l'icône Ajouter une couche Raster.
2. Parcourez le dossier `qgis_sample_data/raster/`, sélectionnez le fichier ERDAS Img `landcover.img` et cliquez sur Ouvrir .
3. Maintenant cliquez sur l'icône Ajouter une couche Vecteur.

4. Parcourez le dossier `qgis_sample_data/gml/`, sélectionnez le fichier GML `lakes.gml` et cliquez sur Ouvrir .
5. Zoomez sur une zone avec quelques lacs.
6. Double-cliquez la couche `lakes` dans liste des cartes pour ouvrir la fenêtre **Propriétés de la couche**.
7. Cliquez sur l'onglet de Convention des signes et sélectionnez le bleu comme couleur de remplissage.
8. Cliquez sur l'onglet étiquettes et cochez la case [X] Afficher les étiquettes.
9. Cliquez sur Appliquer .

FIG. 1: Une session basique de QGIS



Vous pouvez constater combien il est aisé d'afficher des couches raster ou vecteur dans QGIS. Passons aux sections suivantes pour en apprendre plus sur les autres fonctionnalités et paramètres disponibles et la façon de les utiliser.

4 Aperçu des fonctionnalités

Après une première prise en main dans le chapitre 3, nous allons maintenant vous donner un aperçu plus détaillé des fonctionnalités de QGIS. La plupart seront décrites plus précisément dans les chapitres qui leur sont dédiés dans la suite du manuel.

4.1 Démarrer et arrêter QGIS

Dans le chapitre 3.3 vous avez appris comment démarrer QGIS ? Nous allons le répéter ici et vous verrez que QGIS propose des options supplémentaires via la ligne de commande.

– en présumant que QGIS est installé dans le PATH (chemin par défaut), vous pouvez le démarrez en tapant : `qgis` dans une ligne de commande ou en cliquant sur l'icône de raccourci.
– démarrez QGIS en utilisant le menu Démarrer, l'icône de raccourci présent sur le bureau ou encore, en cliquant sur un fichier de projet QGIS
– double-cliquez sur l'icône de votre répertoire Applications.

Pour arrêter QGIS, cliquez sur le menu { Fichier QGIS} > Quitter, ou utilisez le raccourci clavier Ctrl+Q.

4.1.1 Options de ligne de commande

QGIS supporte un certain nombre d'options lorsque démarrer en passant par la ligne de commande. Pour obtenir une liste de ces options, entrez dans votre console `qgis --help`. Le message habituel qui en résulte est :

```
qgis --help
Quantum GIS - 1.0.0 'Kore'
Quantum GIS (QGIS) est un visualisateur de données spatiales, raster ou vecteur.
Usage: qgis [options] [FILES]
  options:
        [--snapshot filename]   emit snapshot of loaded datasets to given file
        [--lang language]       use language for interface text
        [--project projectfile] load the given QGIS project
        [--extent xmin,ymin,xmax,ymax]  set initial map extent
        [--help]                this text

  FILES:
    Files specified on the command line can include rasters,
    vectors, and QGIS project files (.qgs):
     1. Rasters - Supported formats include GeoTiff, DEM
```

```
     and others supported by GDAL
  2. Vectors - Supported formats include ESRI Shapefiles
     and others supported by OGR and PostgreSQL layers using
     the PostGIS extension
```

Astuce 2 EXEMPLE UTILISANT DES OPTIONS DE LIGNE DE COMMANDE
Vous pouvez démarrer QGIS en spécifiant un ou plusieurs fichiers de données. Par exemple, si vous êtes placé dans le répertoire qgis_sample_data vous pouvez démarrer QGIS avec une couche vecteur et un fichier raster dès le démarrage avec la commande suivante : `qgis ./raster/landcover.img ./gml/lakes.gml`

Option `--snapshot`

Cette option permet de créer une capture d'écran de l'affichage courant au format PNG. C'est pratique quand vous avez une longue série de projets et que vous voulez générer un aperçu de vos données. L'image ainsi créée fait 800x600 pixels, un nom de fichier peut être ajouté après `--snapshot`.

Option `--lang`

QGIS se base sur vos paramètres globaux pour définir la langue de l'interface. Si vous voulez en changer, vous devez le spécifier en saisissant un code. Par exemple, `--lang=it` provoquera l'utilisation de la version italienne. Une liste des langues intégrées est visible à `http://wiki.qgis.org/qgiswiki/TranslatorsCorner`

COption `--project`

Démarrer QGIS avec un projet existant est possible, il suffit de rajouter cette option suivie du nom de votre projet et QGIS se lancera avec toutes les couches de ce fichier.

Option `--extent`

Pour démarrer avec une étendue cartographique spécifique, utilisez cette option. Vous devez ajouter les limites de votre étendue dans l'ordre suivant en les séparant par une virgule :

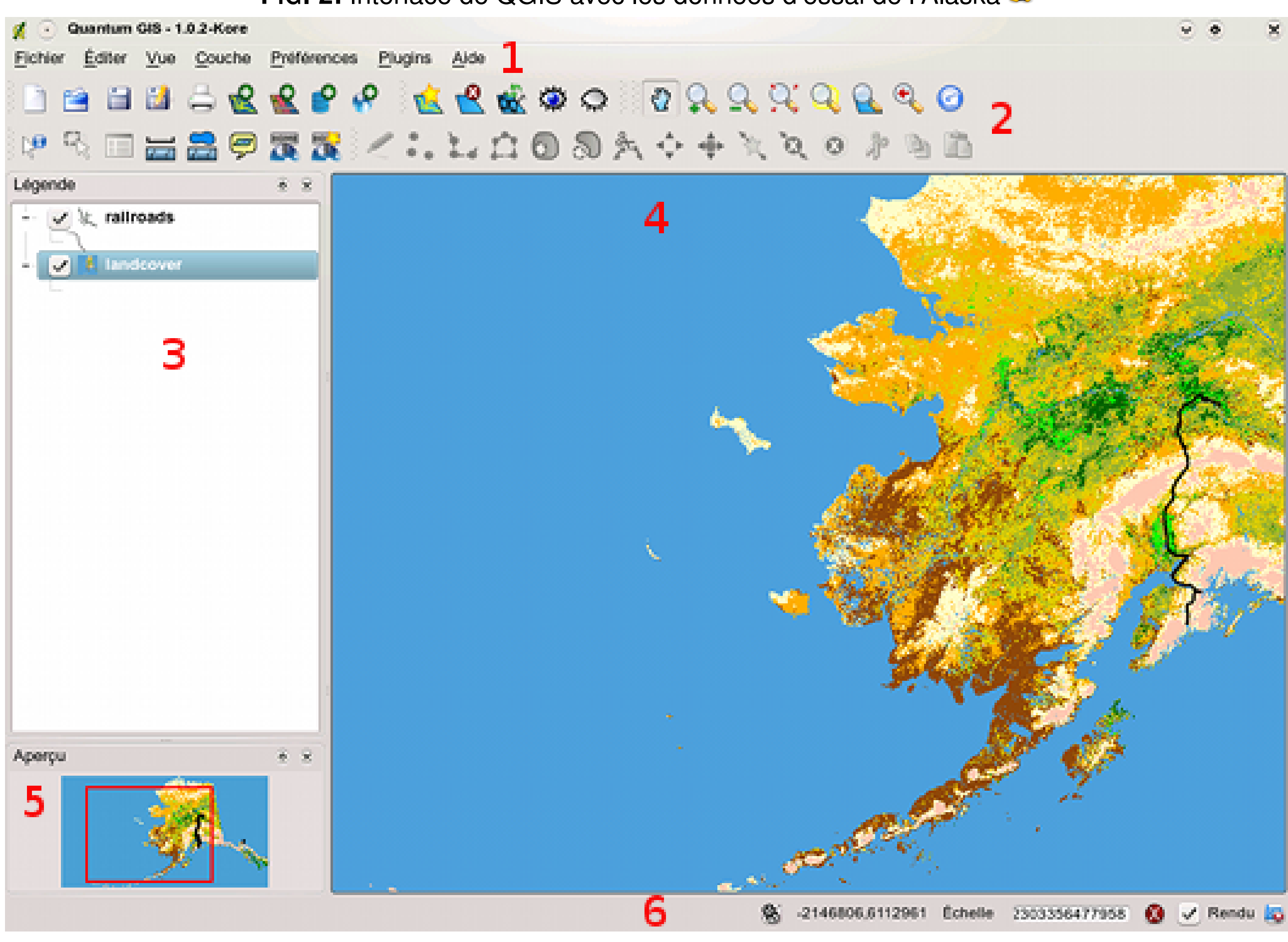
```
--extent xmin,ymin,xmax,ymax
```

4.2 Interface de QGIS

Quand QGIs démarre, l'interface se présente à vous sous la forme affichée ci-dessous (les nombres de 1 à 6 se réfèrent aux six zones majeures de l'interface) :

Note : Les décorations de fenêtre peuvent vous apparaître différemment sur votre système en fonction de votre système d'exploitation et de votre gestionnaire de fenêtre.

FIG. 2: Interface de QGIS avec les données d'essai de l'Alaska



L'interface est divisée en 6 zones distinctes :

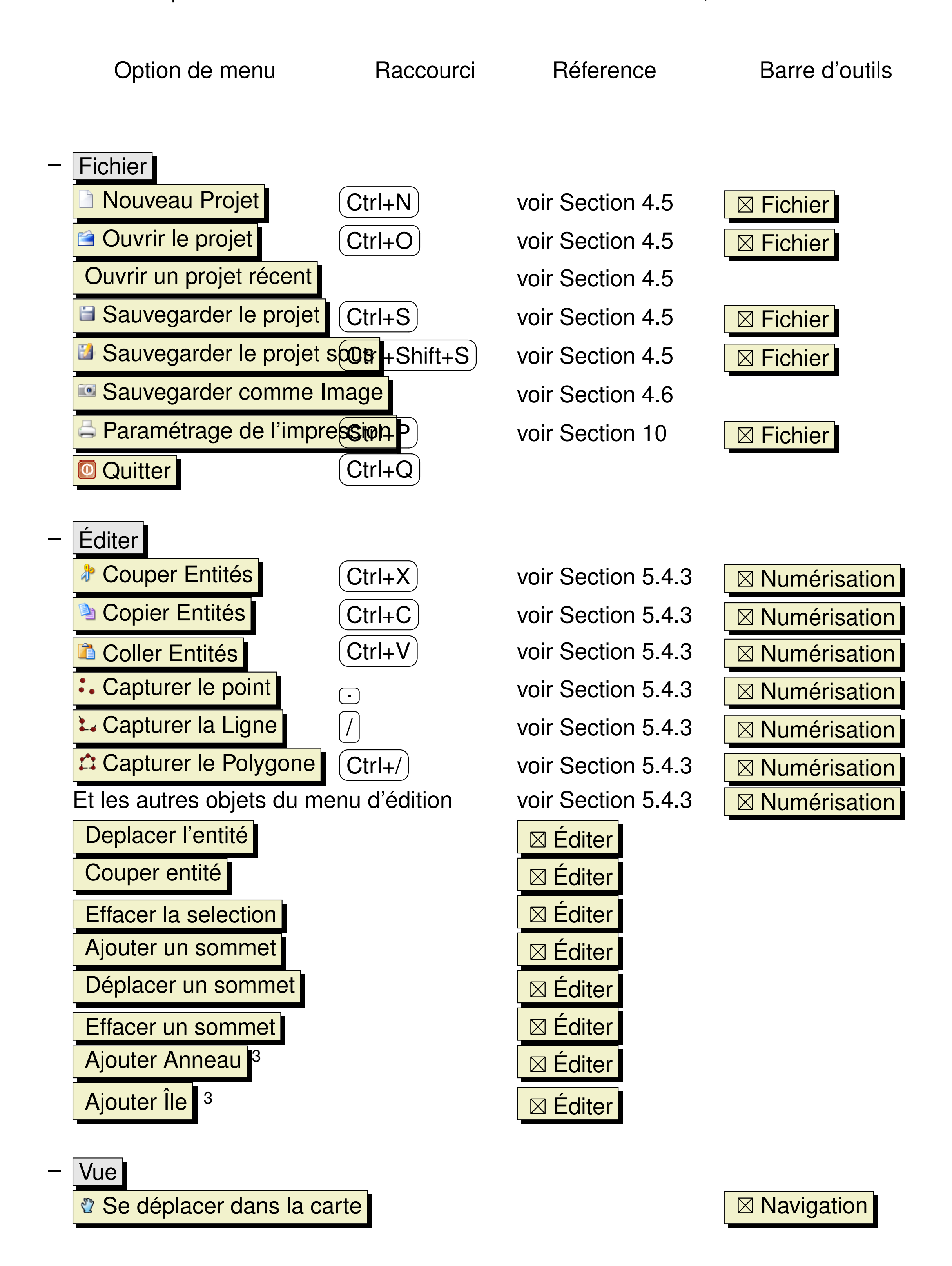
1. Barre de Menu
2. Barre d'Outils
3. Légende de la carte
4. Affichage de la carte
5. Aperçu de la carte
6. Barre de statut

Ces 6 composants sont décrits dans les sections suivantes.

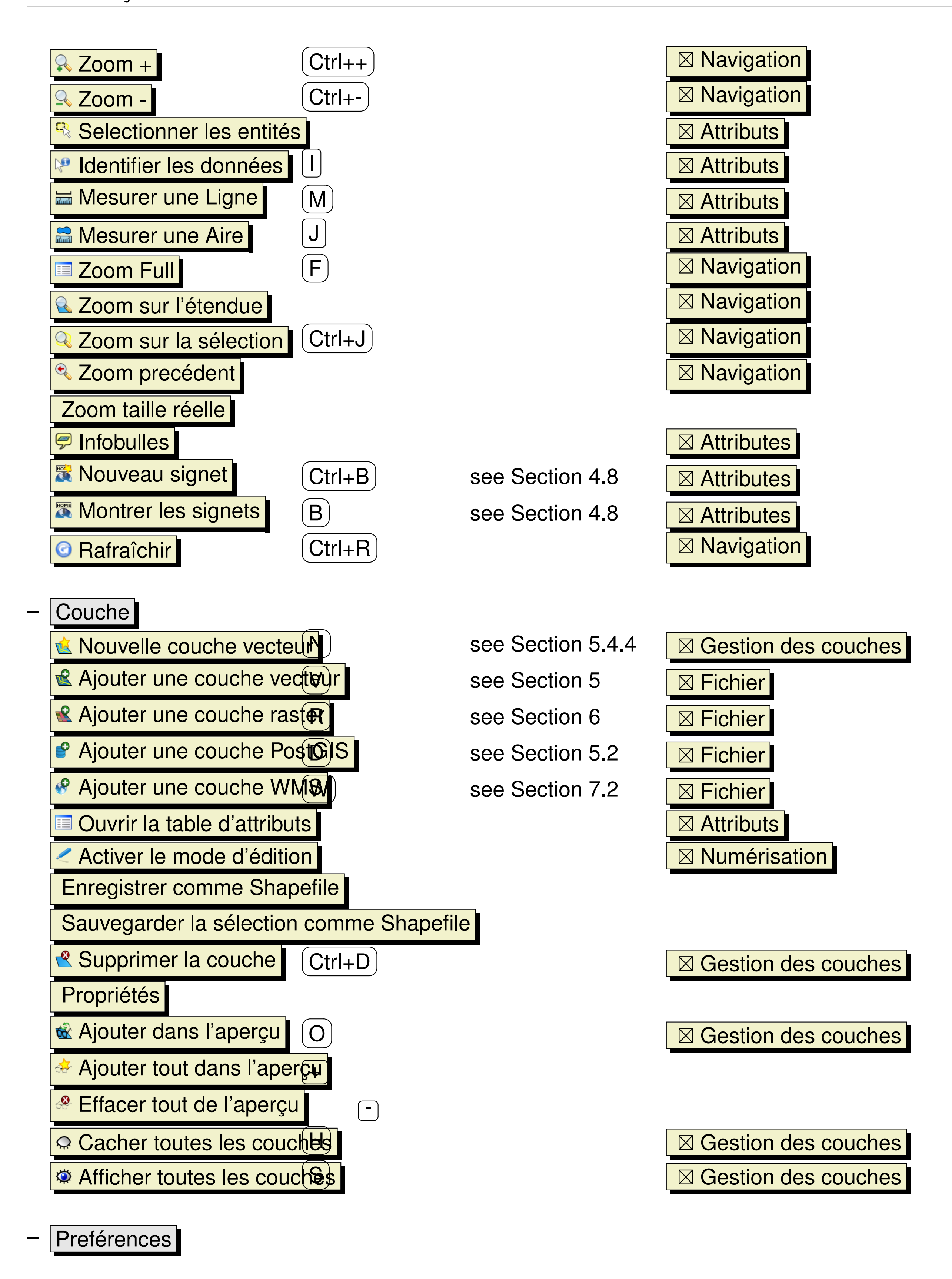
4.2.1 Barre de Menu

La barre de menu fournit un accès aux différentes fonctionnalités de QGIS par le biais de menus hiérarchiques. Le menu supérieur et un résumé de certaines options sont listés ci-dessous, avec les icônes des outils correspondants dans la barre d'outils et leurs raccourcis clavier. Bien que les options

de menu aient des outils qui leur correspondent et vice-versa, les menus ne sont pas organisés comme les barres d'outils. La barre contenant l'outil est affichée à la suite de chaque option de menu. Pour plus d'informations sur les outils et les barres d'outils, veuillez lire la section 4.2.2.



Option de menu	Raccourci	Réference	Barre d'outils
– Fichier			
Nouveau Projet	Ctrl+N	voir Section 4.5	☒ Fichier
Ouvrir le projet	Ctrl+O	voir Section 4.5	☒ Fichier
Ouvrir un projet récent		voir Section 4.5	
Sauvegarder le projet	Ctrl+S	voir Section 4.5	☒ Fichier
Sauvegarder le projet sous	Ctrl+Shift+S	voir Section 4.5	☒ Fichier
Sauvegarder comme Image		voir Section 4.6	
Paramétrage de l'impression	Ctrl+P	voir Section 10	☒ Fichier
Quitter	Ctrl+Q		
– Éditer			
Couper Entités	Ctrl+X	voir Section 5.4.3	☒ Numérisation
Copier Entités	Ctrl+C	voir Section 5.4.3	☒ Numérisation
Coller Entités	Ctrl+V	voir Section 5.4.3	☒ Numérisation
Capturer le point	.	voir Section 5.4.3	☒ Numérisation
Capturer la Ligne	/	voir Section 5.4.3	☒ Numérisation
Capturer le Polygone	Ctrl+/	voir Section 5.4.3	☒ Numérisation
Et les autres objets du menu d'édition		voir Section 5.4.3	☒ Numérisation
Deplacer l'entité		☒ Éditer	
Couper entité		☒ Éditer	
Effacer la selection		☒ Éditer	
Ajouter un sommet		☒ Éditer	
Déplacer un sommet		☒ Éditer	
Effacer un sommet		☒ Éditer	
Ajouter Anneau [3]		☒ Éditer	
Ajouter Île [3]		☒ Éditer	
– Vue			
Se déplacer dans la carte			☒ Navigation

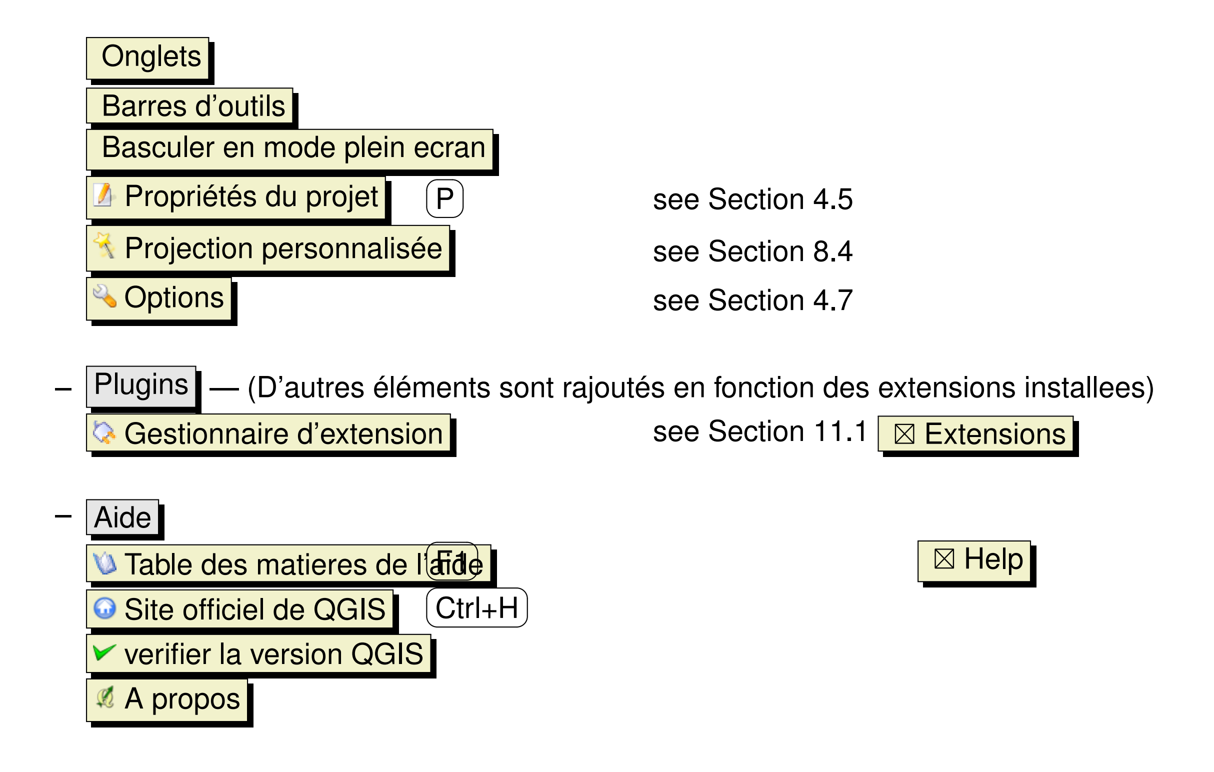


Zoom +	Ctrl++		☒ Navigation
Zoom -	Ctrl+-		☒ Navigation
Selectionner les entités			☒ Attributs
Identifier les données	I		☒ Attributs
Mesurer une Ligne	M		☒ Attributs
Mesurer une Aire	J		☒ Attributs
Zoom Full	F		☒ Navigation
Zoom sur l'étendue			☒ Navigation
Zoom sur la sélection	Ctrl+J		☒ Navigation
Zoom precédent			☒ Navigation
Zoom taille réelle			
Infobulles			☒ Attributes
Nouveau signet	Ctrl+B	see Section 4.8	☒ Attributes
Montrer les signets	B	see Section 4.8	☒ Attributes
Rafraîchir	Ctrl+R		☒ Navigation

– Couche

Nouvelle couche vecteur	N	see Section 5.4.4	☒ Gestion des couches
Ajouter une couche vecteur	V	see Section 5	☒ Fichier
Ajouter une couche raster	R	see Section 6	☒ Fichier
Ajouter une couche PostGIS	D	see Section 5.2	☒ Fichier
Ajouter une couche WMS	W	see Section 7.2	☒ Fichier
Ouvrir la table d'attributs			☒ Attributs
Activer le mode d'édition			☒ Numérisation
Enregistrer comme Shapefile			
Sauvegarder la sélection comme Shapefile			
Supprimer la couche	Ctrl+D		☒ Gestion des couches
Propriétés			
Ajouter dans l'aperçu	O		☒ Gestion des couches
Ajouter tout dans l'aperçu	+		
Effacer tout de l'aperçu	-		
Cacher toutes les couches	H		☒ Gestion des couches
Afficher toutes les couches	S		☒ Gestion des couches

– Preférences



Voir l'annexe **??** pour une description plus aboutie des éléments des menus.

4.2.2 Barre d'outils

La barre d'outils fournit un accès à la majorité des fonctions des menus en plus d'outils additionnels destinés à interagir avec la carte. Chaque outil dispose d'une bulle d'aide qui s'affiche lorsque vous placez votre curseur au-dessus, elle affiche une courte description de son rôle.

Chaque barre de menu peut être déplacée selon vos besoins. Vous pouvez les désactiver en utilisant le bouton droit de votre souris en survolant la barre de menu.

Astuce 3 RESTAURER LA BARRE D'OUTIL

Si vous avez accidentellement masqué toutes vos barres d'outils, vous pouvez les récupérer en sélectionnant Paramétrage > Barre d'outils.

4.2.3 Légende cartographique

La zone de légende cartographique est utilisée pour définir la visibilité et l'ordre d'empilement des couches. Une couche se situant au sommet de la liste de cette légende sera affichée au-dessus de celles qui se situent plus bas dans la liste. La boîte présente à côté de chacune des couches permet d'afficher ou de cacher.

Les couches peuvent être rassemblées en créant un groupe et en y glissant les couches désirées. Pour ce faire, déplacez votre curseur sur la légende, faites un clic droit puis choisissez Ajouter un groupe. Un nouveau dossier est apparu, vous pouvez maintenant glisser et déposer les couches sur le symbole de ce dossier. Il est possible de basculer le mode d'affichage de toutes les couches d'un groupe en décochant seulement le groupe. Pour retirer une couche d'un groupe, il suffit de pointer votre curseur sur elle, de faire un clic droit et de choisir Mette lítem au-dessus. Pour changer le nom du groupe sélectionnez Renommer dans le menu contextuel du groupe.

Le contenu du menu contextuel affiché par un clic droit varie si la couche sélectionnée est un raster ou un vecteur. Pour les couches vectorielles GRASS Basculer en mode édition n'est pas disponible. Veulliez lire la section 9.7 pour plus d'informations sur l'édition de couches vecteurs GRASS.

- **Menu clic-droit pour les couches raster**
  - Zoomer sur lémprise de la couche
  - Zoomer à la meilleure échelle (100%)
  - Montrer dans l'aperçu
  - Effacer
  - Propriétés
  - Renommer
  - Ajouter un groupe
  - Etendre tout
  - Réduire tout
  - Montrer les groupes de fichiers
- **Menu clic-droit pour les couches vecteurs**
  - Zoomer sur lémprise de la couche
  - Montrer dans láperçu
  - Effacer
  - Ouvrir la table d'attributs
  - Basculer en mode édition
  - Sauvegarder comme shapefile
  - Enregistrer la sélection comme shapefile
  - Propriétés
  - Mettre lítem au-dessus
  - Renommer
  - Ajouter un groupe
  - Etendre tout
  - Réduire tout
  - Montrer les groupes de fichiers

- **Menu clic-droit pour les groupes**
  - Effacer
  - Renommer
  - Ajouter un groupe
  - Etendre tout
  - Réduire tout
  - Montrer les groupes de fichiers

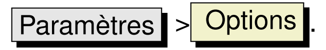
Si plusieurs sources de données vecteurs ont le même type de vecteur (points, lignes ou polygones) et les mêmes attributs, leurs représentations peuvent être groupées. Cela signifie que si la représentation d'une couche est modifiée, toutes les autres en bénéficieront automatiquement. Pour grouper la symbologie, faites un clic-droit dans la zone de légende et sélectionnez Montrer les groupes de fichiers. Les groupes de fichiers relatifs aux couches apparaissent, il est maintenant possible de déplacer un fichier d'un groupe à un autre. SI vous le faites, les fichiers seront regroupés. Notez que QGIS le permet seulement si les 2 couches sont susceptibles de partager le même type de symbologie.

4.2.4 Vue de la carte

C'est la partie centrale de QGIS — les cartes sont affichées dans cette partie ! Celle qui s'affiche dépend des couches raster et vecteurs que vous avez choisi de charger (lire les sections suivantes pour savoir comment charger une couche). La vue de la carte peut être modifiée en portant le focus sur une autre région, ou en zoomant en avant ou en arrière. Plusieurs opérations peuvent être effectuées sur la carte comme il est expliqué dans les descriptions des barres d'outils. La vue de la carte et la légende sont étroitement liées — la carte reflète les changements que vous opérez dans la légende.

Astuce 4 ZOOMER LA CARTE AVEC LA MOLETTE DE LA SOURIS

Vouspouvez utiliser la molette de la souris pour changer le niveau de zoom de la carte. Placez votre curseur dans la zone d'affichage de la carte et faites rouler la molette vers l'avant pour agrandir et faites là rouler vers vous pour zoomer en arrière. La position du curseur est le centre sur lequel va s'opérer le zoom. Vous pouvez modifier le comportement du zoom de la souris en utilisant l'onglet Outils cartographiques dans le menu



> Options.

Astuce 5 DÉPLACER LA CARTE AVEC LES FLÈCHES ET LA BARRE ESPACE

Vous pouvez utiliser les flèches du clavier pour se déplacer sur la carte ? Placez le curseur sur la carte et appuyez sur la flèche droite pour vous diriger vers l'Est, la flèche gauche pour aller vers l'Ouest, la flèche supérieure pour le Nord et la flèche inférieure pour le Sud. Vous pouvez aussi déplacer la carte en gardant la touche espace appuyée et en bougeant la souris.

4.2.5 Aperçu de la carte

La zone d'aperçu de la carte permet d'avoir une vue totale de l'emprise des couches ajoutées au projet. Au sein de cette fenêtre se situe un rectangle qui représente l'étendue de la carte, cela permet de savoir quelle région de la carte vous êtes en train de visualiser. Les étiquettes ne sont pas affichées dans l'aperçu même si les couches visibles ont l'affichage de leurs étiquettes activées. Vous pouvez rajoutez une couche dans l'aperçu en faisant un clic-droit dessus dans la légende ou en sélectionnant [X] Montrer dans l'aperçu. Vous pouvez aussi rajouter des couches ou en ôter de l'aperçu en utilisant les outils d'aperçu dans la barre d'outils.

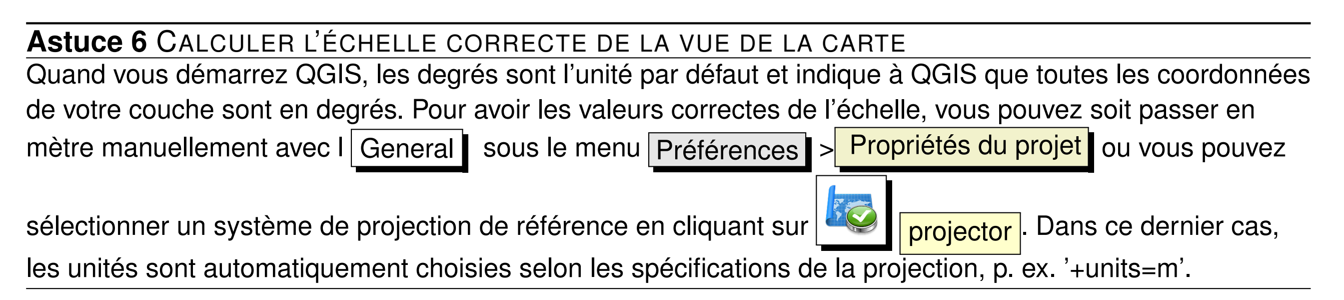
Si vous cliquez et déplacez le rectangle rouge qui montre votre emprise actuelle, la vue principale se mettra en conséquence.

4.2.6 Barre de statuts

La barre de statuts montre votre position dans les coordonnées de la carte (p. ex. mètres ou degrés décimaux) lorsque vous déplacez votre curseur. À gauche de l'affichage des coordonnées se trouve un petit bouton qui bascule entre les coordonnées de la position ou l'étendue de la zone que vous visualisez.

Une barre de progression dans la barre de statuts vous montre le cheminement du rendu au fur et à mesure qu'une couche est dessinée sur l'écran. Dans certains cas, tel que lors de l'établissement de statistiques d'une couche raster, la barre indique le statut des opérations qui prennent du temps.

Si une nouvelle extension ou une mise à jour est disponible, vous verrez un message dans la barre de statut. Sur la droite, une boîte à cocher peut être utilisée pour bloquer le rendu des couches sur la carte (voir Section 4.3). A l'extrémité se situe l'icône de projection, un clic dessus ouvrira la fenêtre de propriétés de projection pour le projet en cours.



Astuce 6 CALCULER L'ÉCHELLE CORRECTE DE LA VUE DE LA CARTE

Quand vous démarrez QGIS, les degrés sont l'unité par défaut et indique à QGIS que toutes les coordonnées de votre couche sont en degrés. Pour avoir les valeurs correctes de l'échelle, vous pouvez soit passer en mètre manuellement avec l General sous le menu Préférences > Propriétés du projet ou vous pouvez sélectionner un système de projection de référence en cliquant sur projector. Dans ce dernier cas, les unités sont automatiquement choisies selon les spécifications de la projection, p. ex. '+units=m'.

4.3 Rendu

Par défaut, QGIS effectue le rendu de toutes les couches visibles à chaque fois que l'affichage de la carte a besoin d'être mise à jour. Les événements qui déclenchent ce rafraîchissement incluent :

– L'ajout d'une couche
– Un déplacement ou un zoom
– Un redimensionnement de la fenêtre de QGIS
– Un changement de la visibilité d'une couche

4.3.1 Rendu dépendant de l'échelle

Le rendu dépendant de l'échelle permet de spécifier l'échelle minimale et maximale à laquelle la couche doit être visible. Pour définir une échelle de rendu, ouvrez la fenêtre de **Propriétées** en double-cliquant sur une couche dans la légende et dans l'onglet Général , saisissez les valeurs voulues et cocher la case [X] Utiliser le rendu dépendant de la mise à l'échelle.

Vous pouvez déterminer les valeurs d'échelle en zoomant au niveau que vous voulez utiliser et en notant les valeurs de la barre d'état.

4.3.2 Contrôler le rendu

Le rendu de la carte peut être contrôlé de différentes manières :

a) Suspendre le rendu

Pour suspendre le rendu, cliquez sur la case [X] Render dans le coin inférieur droit de la barre de statut. Quand cette case n'est pas coché, QGIS ne redessine pas la carte en réponse aux événements décrits dans la section 4.3. Voici quelques cas pour lequel vous pourriez vouloir le faire :

– Ajouter beaucoup de couches et les symboliser avant d'effectuer un rendu potentiellement long
– Ajouter une ou plusieurs couches et définir une échelle
– Ajouter une ou plusieurs couches et et zoomer sur un endroit spécifique
– N'importe quelle combinaison des éléments précédents

Cocher la case [X] Rendu activera de nouveau le rendu et provoquera un rafraîchissement immédiat de la vue active.

b) Définir les options d'ajout de couche

Il est possible de définir une option qui chargera toutes les nouvelles couches sans les dessiner, elles seront ajoutées à la carte, mais la case de visibilité sera décochée par défaut. Pour définir cette option, sélectionnez l'option Préférences > Options et cliquez sur l'onglet Rendu . Décochez la case [X] par défaut les couches supplémentaires sont affichées. Les nouvelles couches ajoutées à la carte seront invisibles par défaut.

Arrêter le rendu

Pour arrêter le rendu de la carte, appuyez sur la touche ESC. Ceci stoppera le rafraîchissement de la vue de la carte et laissera la carte partiellement dessinée. Il est possible qu'il y ait un délai entre le moment où la touche est pressée et le temps que le rendu de la carte soit arrêté.

NOTE : Il n'est pas actuellement possible d'arrêter le rendu de cette manière — ça été désactivé avec le port qt4 du fait d'instabilités.

c) Mettre la jour l'affichage de la carte pendant le rendu

Vous pouvez définir une option pour mettre à jour l'affichage de la carte quand des entités sont dessinées. Par défaut, QGIS n'affiche pas les entités d'une couche tant que la couche entière n'a pas été rendue. Pour mettre à jour l'affichage à mesure que les entités sont lues dans la table attributaire, sélectionnez le menu Settings > Options puis l'onglet Rendu . Mettez comme valeur le nombre d'entités à mettre à jour durant le rendu. Si elle est égale à 0 cela désactive la mise à jour durant le dessin (c'est la valeur par défaut). Une valeur trop basse risque d'impacter les performances, car la vue de la carte sera constamment mise à jour durant la lecture des entités. Il est suggéré de commencer à 500.

d) Influencer la qualité du rendu

Pour influencer la qualité du rendu de la carte vous avez trois possibilités. Dans le menu Préférences > Options puis l'onglet Rendu et sélectionnez/désélectionnez les cases suivantes :

– [X] Les lignes semblent moins déchiquetées aux dépends d'une certaine vitesse d'exécution
– [X] Corriger les polygones remplis de manière erronée
– [X] Rafraîchir en permanence lors du déplacement de la table des matières/carte

4.4 Mesurer

Les mesures fonctionnent uniquement au sein des systèmes de coordonnées projetées (exemple : UTM, Lambert 93). SI la couche active est définie par système géographique de coordonnée (latitude/longitude), les résultats d'une mesure de ligne ou d'aires seront incorrects. Pour y remédier, vous devez mettre un système de coordonnées approprié (voir Section 8).

4.4.1 Mesurer une longueur et une aire

QGIS peut mesurer des distances réelles entre plusieurs points selon un ellipsoïd défini. Pour le configurer, allez dans le menu Préférences > Options puis dans l'onglet Outils cartographiques et choisissez l'ellipsoïde approprié. Cet outil permet de placer des points sur la carte. Chaque longueur de segment s'affiche dans la fenêtre de mesure avec la longueur additionnée totale. Pour stopper les mesures, faites un clic droit.

Les aires peuvent aussi être mesurées.

FIG. 3: Outils de mesure

(a) Mesure de lignes

(b) Mesure d'aires

4.5 les projets

L'état de votre session de QGIS est considéré comme étant un projet. QGIS ne peut travailler que sur un projet à la fois. Les propriétés sont considérées comme étant assignées à un projet ou comme étant par défaut pour les nouveaux projets (voir Section 4.7). QGIS peut enregistrer l'état de votre travail dans un fichier de projet en utilisant le menu the state of your workspace into a project file using the menu options Fichier > Sauvegarder le projet or File > Sauvegarder le projet sous... .

Pour charger un projet dans une session QGIS, aller dans Fichier > Ouvrir un Projet ou Fichier > Ouvrir un projet récent . Si vous souhaitez revenir à une session vierge, aller sur Fichier > Nouveau Projet . Chacune de ces options vous demandera si vous désirez enregistrer le projet si des changements ont été effectués depuis son ouverture ou sa dernière sauvegarde.

Les type d'informations enregistrées dans un projet sont :

- Les couches ajoutées
- Les propriétés des couches ainsi que de symbologie
- La projection de la carte

– L'étendue de la dernière zone de visualisation

Le fichier de projet est enregistré au format XML, il est donc possible de l'éditer en dehors de QGIS si vous savez ce que vous faites. Le format a été modifié à plusieurs reprises depuis les versions antérieures de QGIS, les fichiers enregistrés sous ces dernières peuvent ne plus fonctionner correctement. Pour être averti dans genre de cas, allez dans l'onglet General du menu Préférences > Options et sélectionnez [X] M'avertir lors de l'ouverture d'un fichier projet sauvegardé avec une version précédente de QGIS.

4.6 Sauvegarder l'affichage

Il y a plusieurs façons d'enregistrer un fichier depuis votre session. Nous en avons déjà vu une dans la section 4.5 : sauvegarder dans un fichier de projet. Voici d'autres manières de procéder à l'enregistrement de fichiers :

– Menu option Sauvagarder comme une Image ouvre une fenêtre de dialogue où vous devez saisir le nom, le chemin et le types d'images (PNG ou JPEG).
– Menu option Paramètrage de l'impression ouvre une fenêtre de dialogue où vous pouvez faire une mise en page et imprimer la vue active de la carte (voir Section 10).

4.7 Options d'affichage

Quelques options basiques peuvent être sélectionnés en allant dans le menu Préférences > Options. Les onglets dans lesquels vous pouvez configurer les options sont :

onglet Général

– [X] Demander à sauvegarder les changements apportés au projet si requis
– [X] M'avertir lors de l'ouverture d'un fichier projet sauvegardé avec une version précédente de QGIS
– [X] Changer la couleur de la sélection du fond d'écran
– Changer le thème des icônes
– [X] Mettre les noms de couche en majuscule dans la légende
– [X] Afficher les noms des attributs de classifications dans la légende
– [X] Cacher l'écran de démarrage
– [X] Ouvrir la table d'attributs dans une fenêtre mobile
– Définir le comportement de la table d'attributs (choisir entre montrer toutes les entités, celles sélectionnées et celles présentes dans la vue active)

Onglet de Rendu

- [X] Par défaut les couches supplémentaires sont affichées
- Définir le nombre d’entités à dessiner avant d’actualiser l’affichage.
- [X] Les lignes semblent moins déchiquetées aux dépens d’une certaine vitesse d’exécution
- [X] Corriger les polygones remplis de manière erronée
- [X] Rafraîchir en permanence la table des matières/carte lors du déplacement

Onglet des Outils Cartographiques

- Spécifier le rayon de recherche comme pourcentage de la largeur de la carte)
- Définir l’ellipsoïde pour des calculs de distance
- Définir la couleur de l’étirement pour les outils de mesure
- Définir l’action de la molette de la souris (Zoomer, Zoomer et récentrer, Zoomer sur le curseur de la souris, Rien)
- Définir le facteur de zoom

Onglet de Nuémrisation

- Définir la couleur et la largeur de la ligne d’étirement
- Définir le mode d’accrochage par défaut (à un vertex, un segment, aux vertex et segments)
- Définir la tolérance d’accrochage par défaut (en unités de la couche)
- Définir le rayon de recherche pour l’édition des sommets (en unités de la couche)
- Définir les marqueurs de sommet (croix ou cercle semi-transparent)

Onglet de SCR

- [X] Demander pour le Système de Coordonnées de Référence (SCR)
- [X] La projection globale par défaut du projet qui sera employée.
- [X] La projection par défaut ci-dessous sera employée.
- Sélectionner le Système de Coordonnées de Référence (SCR) par défaut

Onglet de Paramètres du lieu

- [X] Forcer la nationalité du système
- Paramètres de lieu sur votre système

Onglet Proxy

– [X] Utiliser un proxy pour l'accés internet et définition de l'hôte, du port, de l'utilisateur et du mot de passe.

Vous pouvez modifier ces options selon vos besoins, certains de ces changements nécessiteront un redémarrage avant d'être effectifs.

– les paramètres sont enregistrés dans un fichier texte : $HOME/.config/QuantumGIS/qgis.conf
– X les paramètres sont enregistrés dans : $HOME/Library/Preferences/org.qgis.qgis.plist
– les paramètres sont enregistrés dans le registre sous : `\\HKEY\CURRENT\USER\Software\QuantumGIS\qgis`

4.8 Signets spatiaux

Les signets spatiaux vous permettent de marquer une zone de la carte pour y retourner plus tard.

4.8.1 Créer un signet

Pour créer un signet :

1. Déplacez-vous sur la zone concernée
2. Sélectionnez le menu Vue > Nouveau signet ou appuyez sur la touche Ctrl-B.
3. Entrez un nom pour décrire le signet (jusqu'à 255 caractères).
4. Cliquez sur OK pour ajouter le signet ou sur Annuler pour sortir de la fenêtre sans l'enregistrer.

Vous pouvez avoir plusieurs signets portant le même nom.

4.8.2 Travailler avec les signets

Pour utiliser ou gérer les signets allez dans le menu Vue > Montrer les signets. Le dialogue **Signets géospatiaux** vous permet de zoomer ou d'effacer un signet. Vous ne pouvez pas modifier le nom d'un signet ou ses coordonnées.

4.8.3 Zoomer sur un signet

Depuis la fenêtre **Signets géospatiaux**, sélectionner le signet voulu en cliquant dessus puis sur le bouton Zoomer sur . Vous pouvez aussi zoomer en opérant un double-clic.

4.8.4 Effacer un signet

Pour effacer un signet depuis la fenêtre **Signets géospatiaux**, cliquez dessus puis sur le bouton Effacer . Confirmez votre choix en cliquant sur Oui ou annuler en cliquant sur Non .

5 Utiliser des données vecteurs

QGIS gère un grand nombre de formats vecteur, dont ceux gérés par l'extension de conversion de données de la bibliothèque OGR, comme les formats shapefile ESRI, MapInfo MIF (format d'échange) et MapInfo TAB (format natif). Vous trouverez la liste des formats vectoriels supportés par OGR dans l'Annexe A.1.

QGIS gère également les couches PostGIS des bases de données PostgreSQL grâce à l'extension "fournisseur de données" PostgreSQL. La gestion d'autres types de données (par exemple les données texte délimitées) se fait grâce à d'autres extensions "fournisseur de données".

Cette section décrit comment travailler avec deux des formats les plus communs : les shapefiles ESRI et les couches PostGIS. Beaucoup des fonctionnalités de QGIS marchent, de par sa conception, de la même manière, quel que soit le format vecteur des données sources. Il s'agit des fonctionnalités d'identification, de sélection, d'étiquetage et de gestion des attributs.

Le travail sur des couches vectorielles GRASS est décrit dans la Section 9.

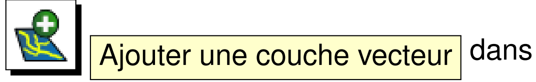
5.1 Shapefiles ESRI

Le format de fichier vecteur standard utilisé par QGIS est le Shapefile ESRI. Il est géré à travers la bibliothèque OGR Simple Feature (`http://www.gdal.org/ogr/`) . Un shapefile correspond en fait à un minimum de trois fichiers :

– `.shp` fichier contenant la géométrie des entités.
– `.dbf` fichier contenant les attributs au format dBase.
– `.shx` fichier d'index.

Dans l'idéal y est associé un autre fichier ayant l'extension `.prj` qui contient les informations sur le système de coordonnées utilisé pour le shapefile. Il peut y avoir encore d'autres fichiers associés aux données shapefile. Si vous souhaitez avoir plus de détails nous vous recommandons de vous reporter aux spécifications techniques du format shapefile, qui se trouve notamment sur `http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf`.

5.1.1 Charger un Shapefile

Pour charger un shapefile, lancer QGIS et cliquez sur 

dans la barre d'outil ou taper simplement V. Ce même outil peut être utilisé pour charger tous les formats gérés par la bibliothèque OGR.

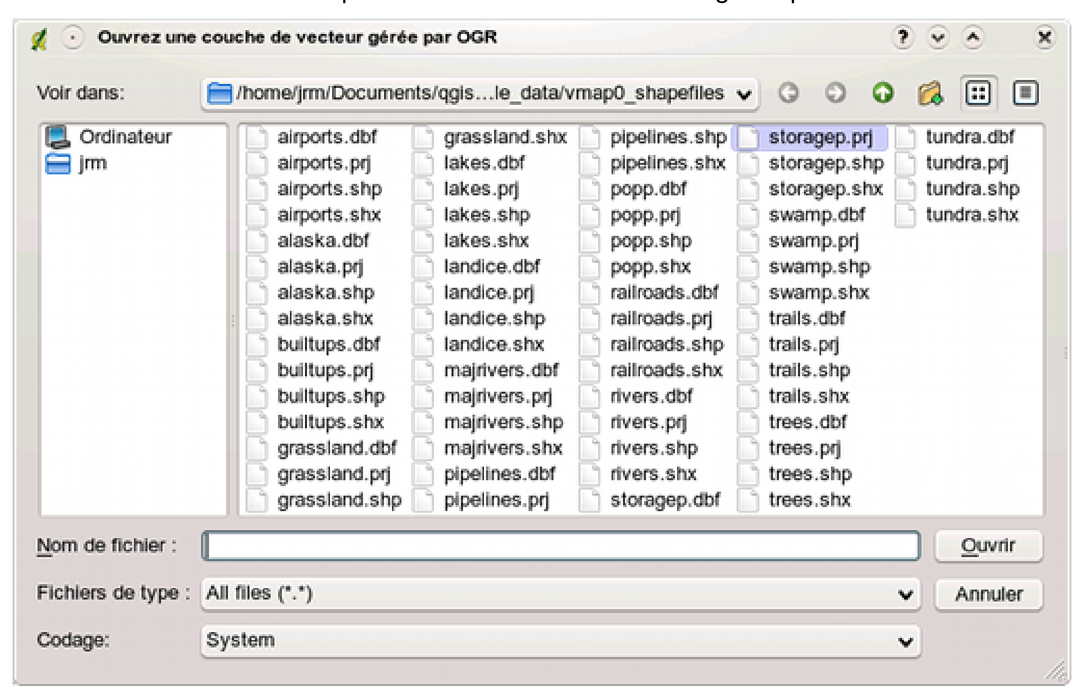
L'outil ouvre alors une fenêtre de dialogue standard (voir Figure 4) qui vous permet de naviguer dans

les répertoires et les fichiers et charger le shapefile ou tout autre format géré. La boîte de sélection Fichiers de type ... vous permet de présélectionner un format de fichier géré par OGR.

Si vous le souhaitez,, vous pouvez également sélectionner le type de codage du shapefile.



FIG. 4: Fenêtre pour ouvrir une couche vecteur gérée par OGR



Sélectionner un shapefile dans la liste puis cliquer sur Ouvrir le charge dans QGIS. La figure 5 montre QGIS après avoir chargé le fichier `alaska.shp`.

Astuce 7 COULEURS DE COUCHES

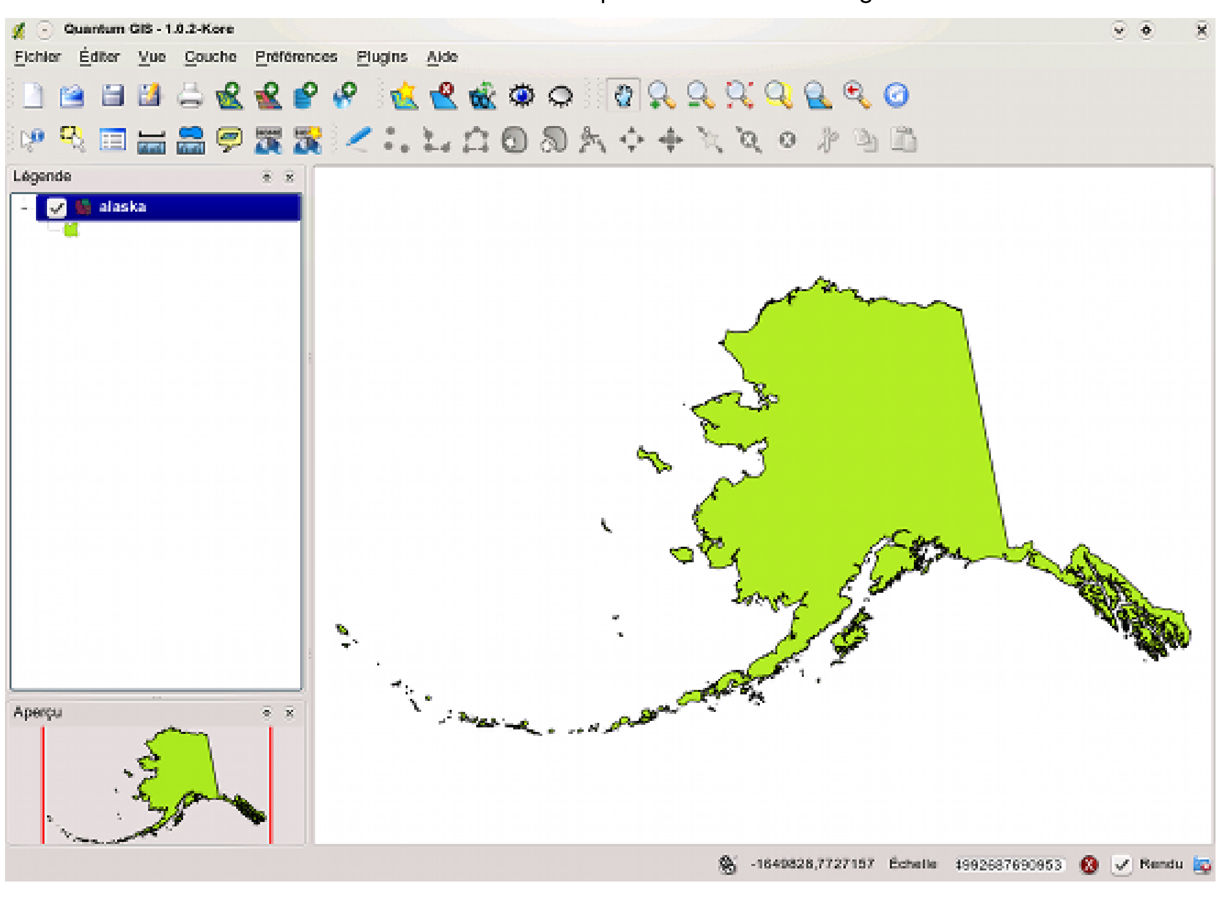
Quand vous ajoutez une couche sur une carte, une couleur aléatoire lui est assignée. En ajoutant plusieurs couches en une fois, différentes couleurs sont assignées à chacune des couches.

Une fois chargée, vous pouvez zoomer sur le shapefile en utilisant les outils de navigation sur la carte. Pour changer la symbologie d'une couche, ouvrez la fenêtre **Propriétés de la Couche** en double-cliquant sur le nom de la couche ou en faisant un clic droit sur son nom dans la légende et en choisissant Propriétés dans le menu qui apparait. Pour plus de détails sur les paramètres de la symbologie des couches vectorielles, référez-vous à la Section 5.3.2.

5.1.2 Améliorer les performances

Pour améliorer les performances de dessin d'un shapefile, vous pouvez créer un index spatial. Un index spatial améliorera à la fois la vitesse d'exécution du zoom et du déplacement panoramique.

FIG. 5: QGIS avec le Shapefile de l'Alaska chargé



Les index spatiaux utilisés par QGIS ont une extension `.qix`.

Voici les étapes de création d'un index spatial :

– Chargez un shapefile
– Ouvrez la fenêtre **Propriétés de la Couche** en double-cliquant sur le nom de la couche dans la légende ou en faisant un clic droit et en choisissant Propriétés dans le menu qui apparait.
– Dans l'onglet Général, cliquez sur le bouton Créez un index spatial.

5.1.3 Charger une couche MapInfo

Pour charger une couche MapInfo, cliquez sur Ajouter une couche vecteur dans la barre d'outils ou tapez V, changez le type de filtre pour Fichiers de type [OGR] MapInfo (*.mif *.tab *.MIF *.TAB) et sélectionnez la couche que

vous souhaitez charger.

5.1.4 Charger une couverture ArcInfo

Charger une couverture ArcInfo se fait de la même manière que pour les couches shapefile ou MapInfo. Cliquez sur Ajouter une couche vecteur dans la barre d'outils ou tapez V pour ouvrir la fenêtre **Ouvrir une couche de vecteur gérée par OGR** et changer le type de filtre pour Fichiers de type All files (*.*). Allez au répertoire de votre couverture et sélectionnez l'un des fichiers suivants (s'ils sont présents pour votre couverture) :

- `.lab` - Pour charger une couche d'étiquettes (polygones d'étiquettes ou points)
- `.cnt` - Pour charger une couche de centroïdes de polygones
- `.arc` - Pour charger une couche d'arcs (lignes)
- `.pal` - Pour charger une couche de polygones.

5.2 Couches PostGIS

Les couches PostGIS sont stockées dans une base de données PostgreSQL. Les avantages de PostGIS sont les possibilités d'indexation spatiale, de filtre et de requête qu'il fournit. En utilisant PostGIS, les fonctions vecteur telles que la sélection ou l'identification fonctionnent avec plus d'exactitude qu'avec les couches OGR dans QGIS.

Pour charger une couche PostGIS, vous devez :

- Dans QGIS, créez une connexion enregistrée à une base de données PostgreSQL (si elle n'a pas été encore défini).
- Connectez-vous à la base de données.
- Sélectionnez la couche à ajouter à la carte.
- En option vous pouvez fournir une clause SQL `where` pour définir les entités de la couche à charger.
- Charger la couche.

5.2.1 Créer une connexion enregistrée

La première fois que utilisez une source de données PostGIS, vous devez créer une connexion à une base de données PostgreSQL qui contient les données. Commencez par cliquer sur le bouton Ajouter une couche PostGIS de la barre d'outils ou sélectionner l'option Ajouter une couche PostGIS... dans le menu Couche ou taper D.

La fenêtre **Ajouter une ou plusieurs tables PostGIS** apparaît. Pour accéder au gestionnaire de connexion, cliquez sur le bouton [Nouveau] pour faire apparaitre la fenêtre **Créer une nouvelle connexion PostGIS**. Les paramètres requis pour la connexion sont présentés dans le tableau1.

TAB. 1: Paramètres de connexion PostGIS

Nom	Un nom pour cette connexion. Il peut être identique à *Base de données*.
Hôte	Nom pour l'hôte de la base de données. Il doit s'agir d'un nom existant, car il sera utilisé pour ouvrir une connexion Telnet ou interroger l'hôte. Si la base de données est sur le même ordinateur que QGIS, mettez simplement 'localhost'.
Base de données	Nom de la base de données.
Port	Numéro de port que le serveur de base de données PostgreSQL écoute. Le port par défaut est 5432.
Nom d'utilisateur	Nom d'utilisateur utilisé pour se connecter à la base de données.
Mot de passe	Mot de passe utilisé avec le *Nom d'utilisateur* pour se connecter à la base de données.

Vous pouvez également activer les options suivantes :

– [X] Sauvegarder le mot de passe
– [X] Uniquement regarder la table geometry_columns
– [X] Uniquement regarder dans le schéma 'public'

Une fois que tous les paramètres et les options sont définis, vous pouvez tester la connexion en cliquant que le bouton [Test de connexion].

Astuce 8 PARAMÈTRES UTILISATEUR DE QGIS ET SÉCURITÉ

Vos paramètres personnalisés pour QGIS sont stockés différemment selon le système d'exploitation. , les paramètres sont stockés dans votre répertoire home dans `.qt/qgisrc`. , les paramètres sont stockés dans la base de registre. Selon votre environnement informatique, stocker vos mots de passe dans vos paramètres QGIS peut présenter des risques vis-à-vis de la sécurité.

5.2.2 Charger une couche PostGIS

Une fois une ou plusieurs connexions définies, vous pouvez charger des couches de la base de données PostgreSQL. Bien sûr, cela nécessite d'avoir des données dans PostgreSQL. Référez-vous à la Section 5.2.4 pour plus de détails concernant l'importation de données dans la base de données.

Pour charger une couche PostGIS, suivez ces étapes :

– Si la fenêtre **Ajouter une ou plusieurs tables PostGIS** n'est pas ouverte, cliquez sur le bouton  Ajouter une couche PostGIS de la barre d'outils.
– Choisissez la connexion dans la liste déroulante et cliquez sur Connecter .
– Trouvez la couche que vous souhaitez ajouter dans la liste des couches disponibles.
– Sélectionnez la en cliquant dessus. Vous pouvez sélectionner plusieurs couches en gradant la touche shift enfoncée quand vous cliquez. Référez-vous à la Section 5.5 pour plus d'informations sur l'utilisation du Constructeur de requête de PostgreSQL pour mieux définir la couche.
– Cliquez sur le bouton Ajouter pour ajouter la couche à la carte.

Astuce 9 COUCHES POSTGIS

Normalement, une couche PostGIS est définie par une entrée dans la table geometry_columns. Depuis la version 0.11.0, QGIS peut charger des couches qui n'ont pas d'entrée dans la table geometry_columns. Ceci concerne aussi bien les tables que les vues. Définir une vue spatiale fournit un moyen puissant pour visualiser vos données. Référez-vous à votre manuel PostgreSQL pour plus d'informations sur la création des vues.

5.2.3 Quelques éléments de détail à propos des couches PostgreSQL

Cette sections contient quelques détails sur la manière dont QGIS accède aux couches PostgreSQL. La plupart du temps, QGIS devrait simplement fournir une liste de tables de base de données qui peuvent être chargées et les charge à la demande. Cependant, si vous avez des problèmes pour charger une table PostgreSQL dans QGIS, les informations données ci-dessous peuvent vous aider à comprendre les messages de QGIS et vous donnez une indication sur comment changer la table ou la vue PostgreSQL pour qu'elle se charge dans QGIS.

QGIS demande que les couches PostgreSQL aient un champ qui peut être utilisé comme clé unique pour la couche. Pour les tables, cela signifie qu'elles doivent avoir une clé primaire ou un champ ayant une contrainte d'unicité. De plus, QGIS impose que cette colonne soit de type int4 (un entier de 4 bites). Si une table ne respecte pas ces conditions, le champ oid sera utilisé à la place. Les performances seront améliorées si le champ est indexé (notez que les clés primaires sont automatiquement indexées dans PostgreSQL).

Si la couche PostgreSQL est une vue, les mêmes conditions s'appliquent mais elles n'ont pas de clé primaire ou de champ ayant une contrainte d'unicité. Dans ce cas, QGIS essayera de trouver un champ de la vue issu d'un champ une table qui convienne. S'il ne peut pas en trouver, QGIS ne chargera pas la couche. Si cela arrive, la solution consiste à modifier la vue de telle sorte qu'elle inclut un champ qui convienne (de type int4 et ayant soit une clé primaire soit une contrainte d'unicité, de préférence indexée).

5.2.4 Importer des données dans PostgreSQL

shp2pgsql

De multiples méthodes existent pour importer des données dans PostgreSQL. PostGIS inclut un utilitaire nommé `shp2pgsql` qui peut être utilisé pour importer des shapefiles dans des bases de données disposant de PostGIS. Par exemple, pour importer le shapefile `lakes.shp` dans une base de données PostgreSQL nommée `gis_data`, utiliser la commande suivante :

```
shp2pgsql -s 2964 lakes.shp lakes_new | psql gis_data
```

Ceci crée une nouvelle couche nommée `lakes_new` dans la base de données usertextgis_data. La nouvelle couche aura l'identifiant de référence spatiale (SRID) 2964. Référez-vous à la Section 8 pour plus d'informations sur les systèmes de référence spatiale et les projections.

Astuce 10 EXPORTER DES JEUX DE DONNÉES DEPUIS POSTGIS
Comme l'outil d'importation `shp2pgsql`, il y a également un outil d'exportation de jeux de données PostGIS en shapefile : `pgsql2shp`. Cet outil est inclus dans la distribution de PostGIS.

Extension SPIT

QGIS est distribué avec une extension nommée SPIT (Shapefile to PostGIS Import Tool). SPIT peut être utilisé pour charger plusieurs shapefiles en une fois et inclut la gestion des schémas. Pour utiliser SPIT, ouvrez le Gestionnaire d'extensions depuis le menu Plugins, cochez la case adjacente à [x] SPIT plugin et cliquez sur OK. L'icône SPIT sera ajoutée à la barre d'outils.

Pour importer un shapefile, cliquez sur le bouton SPIT dans la barre d'outils pour ouvrir la fenêtre **SPIT - Outil d'importation de Shapefile dans PostGIS**. Sélectionnez la base de données à laquelle vous voulez vous connecter et cliquez sur le bouton Connecter. Vous pouvez alors ajouter un ou plusieurs fichiers à la liste en cliquant sur le bouton Ajouter. Pour traiter les fichiers, appuyez sur le bouton OK. La progression de l'importation aussi bien que les erreurs ou les alertes s'afficheront pour chaque shapefile.

Astuce 11 IMPORTER DES SHAPEFILES CONTENANT DES MOTS RÉSERVÉS DE POSTGRESQL
Si un shapefile est ajouté à la liste et que des noms de champs correspondent à des mots réservés dans une base de données PostgreSQL, une fenêtre apparaitra et montrera le statut de chaque champ. Vous pouvez éditer les noms des champs avant l'importation et changer ceux qui correspondent à un mot réservé (ou faire les changements désirés). Toute tentative d'importer un shapefile ayant un champ contenant un mot réservé devrait vraisemblablement échouer.

ogr2ogr

En plus de `shp2pgsql` et `SPIT`, un autre outil est fourni pour importer des données géographiques dans PostGIS : `ogr2ogr`. Il est inclus dans GDAL. Pour importer un shapefile dans PostGIS, tapez la commande suivante :

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres \
password=topsecret" alaska.shp
```

Ceci va importer le shapefile `alaska.shp` dans la base de données PostGIS `postgis` en utilisant l'utilisateur `postgres` avec le mot de passe `topsecret` sur l'hôte *myhost.de*.

Notez qu'OGR doit être compilé avec PostgreSQL pour gérer PostGIS. Vous pouvez vérifier en tapant :

```
ogrinfo --formats | grep -i post
```

Si vous préférez utiliser la commande PostgreSQL `COPY` au lieu de la méthode par défaut, `INSERT INTO`, vous pouvez exporter la variable d'environnement suivante (au moins sur et) :

```
export PG_USE_COPY=YES
```

`ogr2ogr` ne crée pas d'index spatial comme le fait `shp2pgsl`. Vous devez effectuer une étape supplémentaire et le créer manuellement après en utilisant la commande SQL classique `CREATE INDEX` (comme cela est détaillé dans la section suivante 5.2.5).

5.2.5 Améliorer les performances

Récupérer des entités depuis une base de données PostgreSQL peut être long, surtout par un réseau. Vous pouvez améliorer les performances de dessin de couches PostgreSQL en vous assurant qu'un index spatial existe pour chaque couche dans la base de données. PostGIS gère la création d'un index GiST (Generalized Search Tree) pour accélérer les recherches spatiales sur les données.

La syntaxe pour créer un index GiST[4] est la suivante :

```
CREATE INDEX [indexname] ON [tablename]
  USING GIST ( [geometryfield] GIST_GEOMETRY_OPS );
```

[4]les informations de l'index GiST proviennent de la documentation de PostGIS disponible sur `http://postgis.refractions.net`

Notez que pour de grandes tables, créer un index peut prendre du temps. Une fois cet index créé, vous devriez faire une VACUUM ANALYZE. Référez-vous à la documentation de **?** pour plus d'informations.

Voici un exemple de création d'un index GiST :

```
gsherman@madison:~/current$ psql gis_data
Welcome to psql 8.3.0, the PostgreSQL interactive terminal.

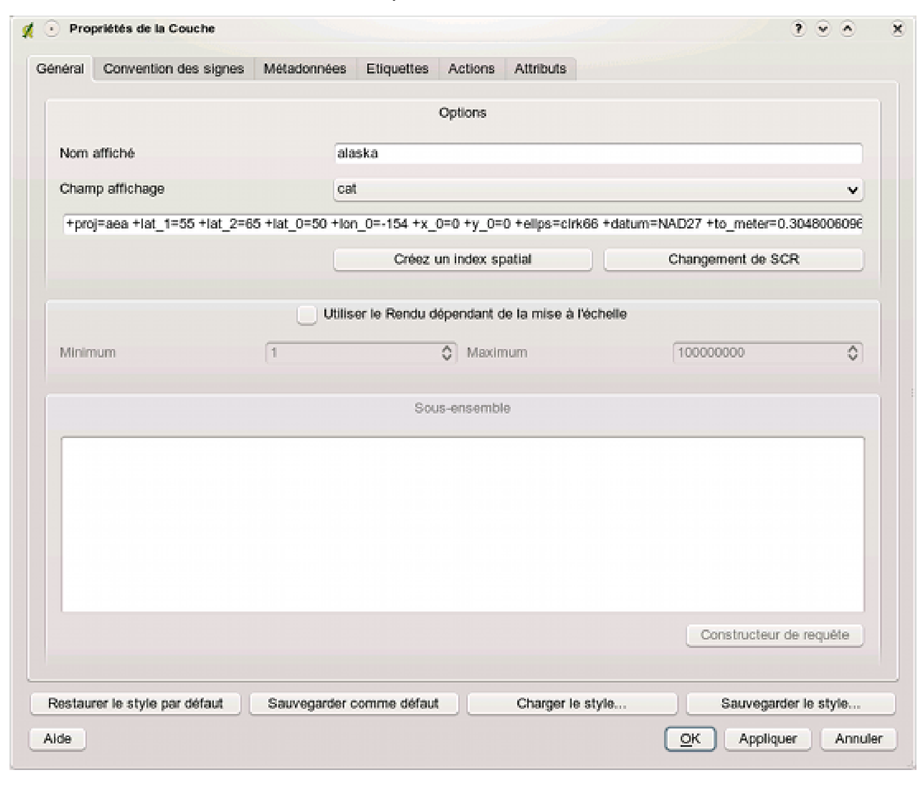
Type:  \copyright for distribution terms
        \h for help with SQL commands
        \? for help with psql commands
        \g or terminate with semicolon to execute query
        \q to quit

gis_data=# CREATE INDEX sidx_alaska_lakes ON alaska_lakes
gis_data-# USING GIST (the_geom GIST_GEOMETRY_OPS);
CREATE INDEX
gis_data=# VACUUM ANALYZE alaska_lakes;
VACUUM
gis_data=# \q
gsherman@madison:~/current$
```

5.3 La fenêtre Propriété des couches vecteur

La fenêtre **Propriétés de la couche** pour une couche vecteur fournit des informations sur la couche, les paramètres de représentation et les options d'étiquetage. Si votre couche a été chargée depuis une base PostgreSQL / PostGIS, vous pouvez également modifier la requête SQL d'appel de la couche, soit manuellement en éditant le SQL dans l'onglet Général soit en appelant la fenêtre **Constructeur de requête** depuis l'onglet Général . Pour accéder à la fenêtre **Propriétés de la couche**, double-cliquez sur la couche dans la légende ou faites un clic droit sur la couche et sélectionnez Propriétés dans le menu qui apparait.

FIG. 6: Fenêtre Propriétés d'une couche vecteur



5.3.1 Onglet Général

L'onglet Général des couches vecteur est très proche de celui des couches raster. Il vous permet de changer le nom affiché, définir des rendus différents selon l'échelle, créer un index spatial du fichier vecteur (uniquement pour les formats gérés par OGR et PostGIS) et visualiser ou changer la projection de la couche.

Le bouton Constructeur de requête vous permet de créer un sous-ensemble d'entité au sein de la couche - mais ce bouton de fonctionne actuellement que lorsque vous ouvrez la table attributaire et cliquez sur le bouton Avancée... .

5.3.2 Onglet Convention des signes

QGIS gère différents types de représentation cartographique pour contrôler la manière pour les entités vectorielles seront affichées. Actuellement, voici les possibilités :

Symbole unique - un style unique est appliqué à tous les objets de la couche.

Symbole gradué - les objets de la couche sont représentés avec des symboles différents selon la valeur qu'ils ont dans un champ définit.

Couleur continue - les objets de la couche sont représentés avec une échelle de couleurs classées selon les valeurs numériques d'un champ définit.

Valeur unique - les objets sont classés par valeur unique dans un champ définit et à chaque valeur correspond un symbole différent.

Pour changer la symbologie d'une couche, double-cliquez simplement dessus dans la légende et la fenêtre de **Propriétés de la couche** apparaîtra.



FIG. 7: Options de symbolisation

(a) Symbole unique

(b) Symbole gradué

(c) Couleur continue

(d) Valeur unique

Options de style

Dans cette fenêtre vous pouvez donner un style à votre couche vecteur. Selon l'option de rendu sélectionnée, vous avez la possibilité de classer vos entités.

Les options de style suivantes s'appliquent quasiment à tous les types de rendus :

Style du contour - style de la ligne qui fait le contour de vos entités. Vous pouvez également le définir à "no pen", pas de contour.

Couleur du contour - couleur du contour de vos entités.

Épaisseur du contour - épaisseur du contour de vos entités.

Couleur de remplissage - couleur de remplissage de vos entités.

Style de remplissage - Style pour le remplissage. En plus des pinceaux proposés, vous pouvez sélectionner [Fill style [? texture ▼]] et cliquez sur le [...] bouton pour sélectionner votre propre style de remplissage. Actuellement, les formats de fichier `*.jpeg, *.xpm et *.png`.

Une fois que vous avez défini le style de votre couche, vous pouvez le sauvegarder dans un fichier séparé (avec l'extension `*.qml`). Pour faire cela, utilisez le bouton [Sauvegarder le style ...]. Inutile de dire que [Charger le style ...] charge vos fichiers sauvegardés.

Si vous voulez utiliser en permanence un style particulier chaque fois que la couche est chargée, utilisez le bouton [Sauvegarder comme défaut] pour en faire le style par défaut. Aussi, si le style ne vous plait pas et que vous le modifiez, utilisez le bouton [Restaurer le style par défaut] pour en faire votre style par défaut.

Transparence d'une couche vecteur

QGIS 1.0.0 permet de définir une transparence pour chaque couche vecteur. Ceci peut-être fait avec le curseur [Transparence 0% ▽▬▬▬] de l'onglet [Convention des signes] (voir fig. 6). Ceci est très utile pour superposer plusieurs couches vecteur.

5.3.3 Onglet Métadadonnées

L'onglet [Métadadonnées] contient les informations sur la couche dont le type et la localisation, le nombre d'entités, le type des entités et les possibilités d'éditions. Les sections Système spatial de référence de la couche qui fournit les informations sur la projection et Information de champ d'attribut qui liste les champs et leur type sont affichées dans cet onglet. Cet onglet constitue un moyen rapide d'obtenir des informations sur une couche.

5.3.4 Onglet Étiquettes

L'onglet [Étiquettes] vous permet d'activer la fonctionnalité d'étiquetage et de gérer un certain nombre d'options liées à la police de caractère, au placement, au style, à l'alignement et au buffering.

Nous allons illustrer tout cela en étiquetant le shapefile des lacs du jeu de données `qgis_example_dataset` :

1. Charger le shapefile `alaska.shp` et le fichier GML `lakes.gml` dans QGIS.
2. Zoommez légèrement sur votre coin préféré avec quelques lacs.
3. Rendez active la couche `lakes`
4. Ouvrez la fenêtre **Propriétés de la couche**.
5. Cliquez sur l'onglet Étiquettes .
6. Cochez la case [X] Afficher les étiquettes pour activer l'étiquetage.
7. Choisissez le champ à utiliser pour les étiquettes. Ici, nous utiliserons le Field containing label NAMES ▼.
8. Choisissez un libellé par défaut pour les lacs n'ayant pas de nom. Ce libellé sera utilisé chaque fois que QGIS rencontre un lac n'ayant pas de valeur dans le champ NAMES.
9. Cliquez sur Appliquer .

Maintenant, nous avons des étiquettes. De quoi ont-elles l'air ? Elles sont probablement trop grandes et mal placées par rapport au symbole marqueur des lacs.

Sélectionnez l'entrée Police et utilisez les boutons Police et Couleur pour définir la police et la couleur. Vous pouvez également changer l'angle et le placement de l'étiquette.

Pour changer la position du texte par rapport à l'entité :

1. Cliquez sur l'entrée Police
2. Changer le placement en sélectionnant l'un des boutons radio dans le groupe **Placement**. Pour corriger nos étiquettes, choisissez le bouton radio ⊙ Droite.
3. La **Taille de la police des unités** vous permet de choisir entre des ⊙ Points ou des ⊙ Unités de carte
4. Cliquez sur Appliquer pour visualiser les changements sans fermer la fenêtre.

Ca à l'air plus joli mais les étiquettes sont encore trop proches des marqueurs. Pour corriger cela, nous pouvons utiliser les options de l'entrée Position . Ici, nous pouvons ajouter un décalage dans les directions X et Y. Ajouter un décalage de 5 en X déplacera vos étiquettes et les rendra plus lisibles. Bien sûr si vos symboles marqueurs ou votre police sont plus grands un décalage plus important sera nécessaire.

Un dernier ajustement reste à faire sur les étiquettes : un buffer . Il s'agit de créer un fond autour des étiquettes pour les faire mieux ressortir. Pour faire un buffer sur les étiquettes des lacs :

1. Cliquez sur l'entrée Buffer .
2. Cliquez sur la case à cocher X Buffer Labels ? pour activer le buffer.
3. Choisissez une taille de buffer en utilisant les flèches.
4. Choisissez une couleur en cliquant sur Couleur puis choisissez votre couleur favorite grâce au sélecteur. Si vous le souhaitez, vous pouvez également ajouter un peu de transparence au buffer.
5. Cliquez sur Appliquer pour voir si les changements vous plaisent.

Si le résultat ne vous plaît pas, ajustez les paramètres et re-testez en cliquant sur Appliquer .

Un buffer d'un point semble donner un bon résultat. Notez que vous pouvez également spécifier une taille de buffer en unités de la carte si cela vous convient mieux.

Les autres entrées de l'onglet Étiquettes vous permettent de contrôler l'apparence des étiquettes en utilisant les attributs stockés dans la couche. Les entrées commençant par Data defined vous permettent de définir tous les paramètres des étiquettes en utilisant des champs de la couche.

Notez que l'onglet Étiquettes propose une **Prévisualisation** montrant une de vos étiquettes.

5.3.5 Onglet Actions

QGIS est capable d'effectuer des actions basées sur les attributs d'une entité. Il peut s'agir de nombreuses actions, par exemple exécuter un programme avec des arguments construits à partir des attributs d'une entité, ou encore, passer des paramètres à un outil de publication de rapports sur internet.

Les actions sont utiles si vous voulez exécuter fréquemment une application externe ou charger une page web basée sur une ou plusieurs valeurs de votre couche vecteur. Un exemple d'application serait d'effectuer une recherche basée sur une valeur d'attribut. C'est l'idée utilisée dans les paragraphes qui suivent.

Définir des actions

Les actions sur les attributs sont définies dans la fenêtre **Propriétés de la couche** des couches vecteur. Pour définir une action, ouvrez la fenêtre de **Propriétés de la couche** et cliquez sur l'onglet Actions . Donnez un nom descriptif à l'action. L'action elle-même doit contenir le nom de l'application qui sera exécutée quand l'action sera invoquée. Vous pouvez ajouter un ou plusieurs champs d'attributs comme argument pour l'application. Quand l'action est invoquée n'importe quelle chaîne de caractère précédée de % et correspondant au nom d'un champ sera remplacé par la valeur de ce champ. Le caractère spécial %% sera remplacé par la valeur d'un champ qui a été sélectionné

par le résultat d'un Identifier ou dans la table d'attributs (voir Utiliser les actions, ci-dessous). Des guillemets peuvent être utilisés pour grouper du texte en un seul argument pour le programme, le script ou la commande. Les guillemets seront ignorés s'ils sont précédés d'un antislash.

Si vous avez des noms de champs qui sont contenus dans d'autres noms de champs (par exemple, `col1` et `col10`), vous devez l'indiquer en entourant le nom de champ (le caractère %) par des crochets (par exemple `[%col10]`). Ceci évitera de prendre le nom de champ `%col10` pour `%col1` avec un `0` à la fin. Les crochets seront retirés quand QGIS substituera le nom par la valeur du champ. Si vous voulez que le champ à substituer soit entouré de crochets, utilisez un deuxième jeu de crochets comme ici : `[[%col10]]`.

La fenêtre **Résultats identifiés** inclue une entrée *(Dérivé)* qui contient des informations pertinentes selon le type de couche. Les valeurs de cette entrée sont accessibles de la même manière que les autres champs en ajoutant `(Derived).` avant le nom du champ. Par exemple, une couche de points à un champ `X` et `Y` et leur valeur peut être utilisée dans l'action avec `%(Derived).X` et `%(Derived).Y`. Les attributs dérivés sont disponibles uniquement depuis la fenêtre **Résultats identifiés** et pas la **Table d'attributs**.

Deux exemples d'action sont proposés ci-dessous :

- `konqueror http ://www.google.com/search ?q=%nam`
- `konqueror http ://www.google.com/search ?q=%%`

Dans le premier exemple, le navigateur internet konqueror est lancé avec une URL. L'URL effectue une recherche Google sur la valeur du champ `nam` de la couche vecteur. Notez que l'application ou le script appelé par l'action doit être dans le path sinon vous devez fournir le chemin complet vers l'application. Pour être certain, nous pouvons réécrire le premier exemple de cette manière : `/opt/kde3/bin/konqueror http ://www.google.com/search ?q=%nam`. Ceci assurera que l'application konqueror sera exécutée quand l'action sera invoquée.

Le deuxième exemple utilise la notation %% dont la valeur ne dépend pas d'un champ en particulier. Quand l'action est invoquée, %% sera remplacé par la valeur du champ sélectionné dans les résultats de l'identification ou dans la table d'attributs.

Utiliser les actions

Les actions peuvent être invoquées soit depuis la fenêtre **Résultats identifiés** soit depuis la **Table d'attributs**. (Rappelez-vous que ces fenêtres s'ouvrent en cliquant sur Identifier les données ou Ouvrir la table d'attributs.) Pour invoquer une action, faites un clic-droit sur un enregistrement et choisissez l'action depuis le menu qui apparaît. Les actions sont listées dans le menu par le nom que vous leur avez donné en les définissant. Cliquez ensuite sur l'action que vous souhaitez invoquer.

Si vous invoquez une action qui utilise la notation %%, faites un clic droit sur la valeur du champ que vous souhaitez passer en argument à l'application ou au script dans la fenêtre **Résultats identifiés** ou la **Table d'attributs**.

Voici un autre exemple qui récupère des données d'une couche vecteur et qui les insère dans un fichier utilisant bash et la commande `echo` (cela ne marchera que sur et peut-être). La couche en question à des champs pour le nom d'espèce `taxon_name`, la latitude `lat` et la longitude `long`. Je souhaiterais faire une sélection spatiale des localités et exporter ces valeurs des enregistrements sélectionnés dans un fichier texte (ils apparaissent en jaune sur la carte dans QGIS). Voici l'action qui permettra de le faire :

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

Après avoir sélectionné quelques localités et lancé l'action sur chacune, le fichier de destination ressemblera à ça :

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

Comme exercice, nous allons créer une action qui réalise une recherche Google sur la couche `lakes`. Tout d'abord, nous avons besoin de déterminer l'URL nécessaire pour effectuer une recherche sur un mot clé. Il suffit simplement d'aller sur Google et faire une recherche simple puis récupérer l'URL dans la barre d'adresse de votre navigateur. De cela, nous en déduisons la formulation : `http://google.com/search?q=qgis`, où `qgis` est le terme recherché. À partir de tout cela, nous pouvons poursuivre :

1. Assurez-vous que la couche `lakes` est chargée.
2. Ouvrez la fenêtre **Propriétés de la couche** en double cliquant sur la couche dans la légende ou en faisant un clic-droit et en choisissant Propriétés dans le menu qui apparaît.
3. Cliquez sur l'onglet Actions .
4. Entrez un nom pour l'action, par exemple `Recherche Google`
5. Pour l'action, nous devons fournir le nom du programme externe à lancer. Dans ce cas, nous allons utiliser Firefox. Si le programme n'est pas dans votre path, vous devez fournir le chemin complet.
6. A la suite du nom de l'application externe, ajoutez l'URL utilisée pour faire la recherche Google, jusqu'au terme de recherche mais sans l'ajouter : `http://google.com/search?q=`

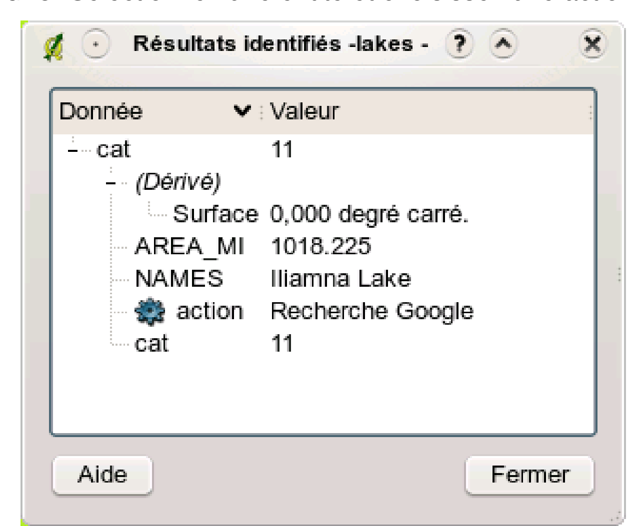
7. Le texte dans le champ Action devrait ressembler à ça :
   `firefox http://google.com/search?q=`
8. Cliquez sur le menu déroulant contenant les noms des champs pour la couche `lakes`. Il est situé juste à gauche du bouton [Insérer un champ].
9. Dans le menu déroulant, sélectionnez [NAMES ▼] et cliquez sur [Insérer un champ].
10. Le texte de votre action devrait maintenant ressembler à ça :
    `firefox http://google.com/search?q=%NAMES`
11. Pour finaliser l'action, cliquez sur le bouton [Insérer une action] ...

L'action est donc entièrement définie et prête à être utilisée. Le texte final de l'action devrait correspondre à ça :

```
firefox http://google.com/search?q=%NAMES
```

Nous pouvons maintenant utiliser l'action. Fermer la fenêtre **Propriétés de la couche** et zoomez sur une zone d'intérêt. Assurez-vous que la couche `lakes` est active puis identifiez un lac. Dans la fenêtre de résultats, vous constatez que notre action est maintenant visible :

FIG. 8: Sélectionnez une entité et choisissez une action



Quand vous cliquez sur l'action, cela ouvre Firefox et charge l'URL `http://www.google.com/search?q=Tustumena`. Il est également possible d'ajouter d'autres champs attributs à l'action. Pour faire cela, vous pouvez ajouter un "+" à la fin du texte de l'action, sélectionnez un autre champ et cliquez sur [Insérer un champ]. Dans cet exemple, la recherche sur un autre champ n'aurait pas de sens.

Vous pouvez définir de multiples actions pour une couche et chacune apparaitra dans la fenêtre **Résultats identifiés**. Vous pouvez également invoquer des actions depuis la table d'attributs en sélectionnant une colonne et en faisant un clic-droit puis en choisissant l'action dans le menu qui apparaît.

Vous pouvez imaginer toute sorte d'utilisations pour ces actions. Par exemple, si vous avait une couche de points contenant la localisation d'images ou de photos ainsi qu'un nom de fichier, vous pouvez créer une action qui lancera un visualisateur pour afficher les images. Vous pouvez également utiliser les actions pour lancer des rapports sur internet pour un champ attributaire ou une combinaison de champs, en les spécifiant de la même manière que pour une recherche

5.3.6 Onglet attributs

Dans l'onglet Attributs, il est possible de manipuler les attributs du jeu de données sélectionné. Les boutons Ajouter une colonne et Supprimer une colonne peuvent être utilisés lorsque le jeu de données est en mode édition. A ce moment les colonnes des couches PostGIS seulement peuvent être éditées car cette fonctionnalité n'est pas encore supportée dans la bibliothèque OGR.

Le bouton Basculer en mode édition permet de passer dans ce mode.

widget d'édition

Dans l'onglet Attributs vous trouverez un colonne `Editer le bidule` et une colonne `valeur`. Ces deux colonnes peuvent être utilisées pour définir les valeurs ou les plages de valeurs permises lors de l'ajout d'attributs dans une colonne. Elles sont utilisées pour générer différents widgets d'édition dans la fenêtre des attributs. Ces widgets sont :

– édition de ligne : un champ d'édition qui permet d'entrer du texte simple (ou de restreindre à des nombres pour des attributs de type numériques).
– valeurs uniques : une liste de valeurs d'attribut uniques de toutes les entités pré-existantes et présentées dans la liste déroulante pour sélection.
– valeur unique (éditable) : une combinaison de 'édition de ligne' et 'valeurs uniques'. Le champ édité autorise les nouvelles valeurs et les ajoute à la liste des valeurs uniques.
– Palette de valeur : une liste déroulante pour sélectionner une valeur dans une liste spécifiée dans le champ `valeurs` de l'onglet Attributs. Les valeurs possibles sont délimitées par un point virgule (par exemple `fort;moyen;faible`). Il est également possible d'associer une étiquette à chaque valeur, qui sera délimitée par un signe égal (par exemple `fort=1;moyen=2;faible=3`). L'étiquette sera visible dans la liste déroulante à la place de la valeur.
– classification : si un rendu de type valeur unique est choisi pour la couche, ces valeurs seront utilisées pour les classes proposées pour la sélection dans la liste déroulante.
– Domaine de validité (éditable) : un champ d'édition qui permet de restreindre les valeurs numériques à une plage donnée. Cette plage est spécifiée en entrant les valeurs minimum et un

maximum délimitées par un point virgule (par exemple `0;360`) dans le champ `valeurs` de l'onglet Attributs .

– Domaine de validité (barre de défilement) : une barre de défilement permet de sélectionner une valeur dans une plage et avec une précision données. La plage est spécifiée par des valeurs minimum et maximum et un pas (par exemple `0;360;10`) dans le champ `valeurs` de l'onglet Attributs .
– nom de fichier : le widget d'édition de ligne est accompagné d'un bouton. Pressé, il permet de sélectionner un nom de fichier grâce à une fenêtre standard d'exploration des fichiers.

5.4 Éditer

Les capacités d'édition de QGIS sur les géométries vecteur sont basiques. Avant d'aller plus loin, notez que la gestion de l'édition dans QGIS reste encore préliminaire. Avant d'effectuer des éditions, créez toujours une sauvegarde du jeu de données que vous allez éditer.

Note - la procédure pour éditer des couches GRASS est différente - voir Section 9.7 pour plus de détails.

5.4.1 Définir le rayon de tolérance d'accrochage et de recherche

Avant de pouvoir éditer des sommets, il est très important de fixer la tolérance d'accrochage et le rayon de recherche à des valeurs qui nous permettent d'éditer les géométries vecteur de manière optimale.

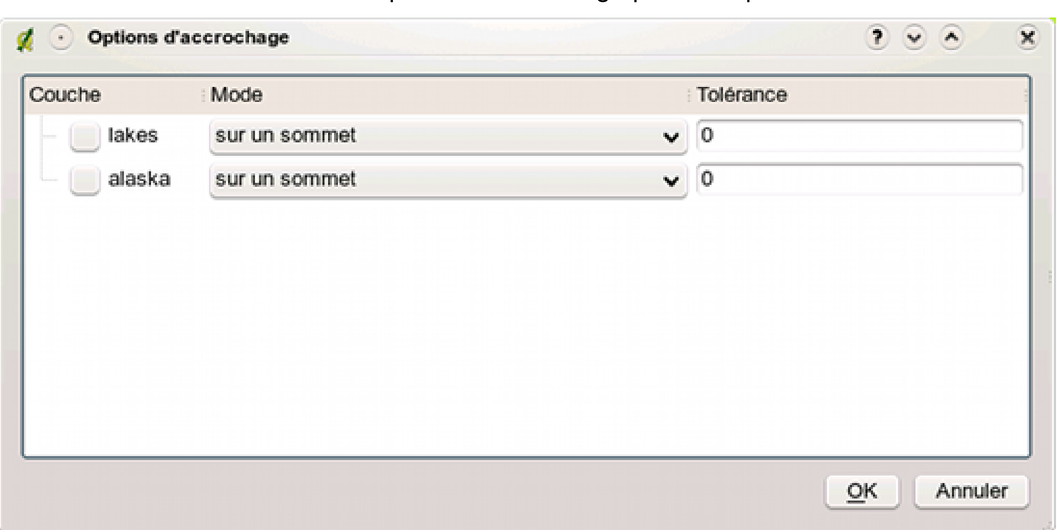
Tolérance d'accrochage

La tolérance d'accrochage est la distance que QGIS utilise pour `chercher` le sommet ou le segment le plus près que vous cherchez à connecter lorsque vous créez un nouveau sommet ou en déplacez un existant. Si vous n'êtes pas dans la tolérance d'accrochage, QGIS va laisser le vertex à l'endroit où vous lâchez le bouton de la souris, au lieu de l'accrocher à un sommet ou un segment existant.

1. Une tolérance générale, commune à tout le projet, peut-être définie dans Préférences > Options. Dans l'onglet Numérisation , vous pouvez choisir le mode d'accrochage par défaut : sur un sommet, sur un segment ou sur un sommet ou un segment. Vous pouvez également définir une tolérance d'accrochage par défaut et un rayon de recherche pour les éditions de sommets. Rappelez-vous que la tolérance est dans l'unité de la couche. Dans notre projet de numérisation (travail sur le jeu de données Alaska), les unités sont en pieds. Le résultat peut varier mais une tolérance de l'ordre de 300 pieds devrait être convenable à une échelle de 1 :10 000.
2. Une tolérance d'accrochage liée à une couche peut être définie dans Préférences > Propriétés du projet.... Dans l'onglet Général , section **Numériser**, vous pouvez cliquer

sur Options d'accrochage... pour activer et ajuster le mode d'accrochage et la tolérance pour chaque couche (voir Figure 9).

FIG. 9: Édition des options d'accrochage pour chaque couche



Rayon de recherche

Le rayon de recherche est la distance que QGIS utilise pour `chercher` le sommet le plus proche que vous souhaitez déplacer quand vous cliquez sur la carte. Si vous n'êtes pas dans le rayon de recherche, QGIS ne trouvera ni ne sélectionnera de sommet à éditer et une fenêtre d'alerte désagréable apparaitra. La tolérance d'accrochage et le rayon de recherche sont définis dans les unités de la carte, vous allez peut-être avoir besoin d'expérimenter différentes valeurs avant de trouver la bonne. Si vous spécifiez une tolérance trop grande, QGIS risque d'accrocher le mauvais sommet, surtout si vous avez un grand nombre de sommets à proximité. Définissez un rayon de recherche trop petit et QGIS ne trouvera rien à déplacer.

Le rayon de recherche pour l'édition des sommets dans l'unité de la couche peut être défini dans l'onglet Numérisation de Préférences > Options. Au même endroit que vous définissez la tolérance d'accrochage pour tout le projet.

5.4.2 Édition topologique

En plus des options d'accrochage pour chaque couche, l'onglet Général du menu Préférences > Propriétés du projet... propose quelques fonctionnalités topologiques. Dans le groupe d'options de Numérisation, vous pouvez [X] Activer l'édition topologique et/ou activer

[X] Éviter les intersections de nouveaux polygones.

Activer l'édition topologique

L'option [X] Activer l'édition topologique permet d'éditer en gardant des limites communes entre les polygones. QGIS "détecte" une limite commune entre les polygones et vous avez simplement à déplacer le sommet une fois et QGIS s'occupera de mettre à jour l'autre limite.

Éviter les intersections de nouveaux polygones

La deuxième option topologique, [X] Éviter les intersections de nouveaux polygones, permet d'éviter des recouvrements entre les polygones. Cela permet de numériser des polygones adjacents plus rapidement. Si vous avez déjà un polygone, avec cette option, vous pouvez numériser le second de manière à ce qu'ils intersectent et QGIS coupera le second polygone aux limites communes. L'avantage est que les utilisateurs n'ont pas à numériser tous les sommets des limites communes.

5.4.3 Édition d'une couche existante

Par défaut, QGIS charge les couches en lecture seule : c'est une sécurité pour éviter d'éditer accidentellement une couche si la souris a glissé. Cependant, vous pouvez choisir d'éditer une couche du moment que le fournisseur de données le gère et que la source de données est éditable (i.e. fichiers qui ne sont pas en lecture seule).

L'édition d'une couche est plus flexible lorsqu'il s'agit de sources de données PostgreSQL/PostGIS.

Astuce 12 INTÉGRITÉ DES DONNÉES

Sauvegarder vos données avant de se lancer dans une édition est toujours une bonne idée. Bien que les auteurs de QGIS ont fait beaucoup d'efforts pour préserver l'intégrité de vos données, nous n'offrons aucune garantie.

Astuce 13 MANIPULATION DES DONNÉES ATTRIBUTAIRES

Actuellement, seules les couches PostGIS gèrent l'ajout ou la suppression de champs attributaires dans cette fenêtre. Dans les versions futures de QGIS, d'autres sources de données seront gérées puisque cette fonctionnalité a récemment été ajoutée à GDAL/OGR > 1.6.0

Astuce 14 FRÉQUENCE DE SAUVEGARDE

N'oubliez pas de cliquer sur Basculer en mode édition régulièrement. Cela vous permet de sauvegarder les changements récents mais également de confirmer que votre source de données accepte tous vos changements.

Toute session d'édition commence par un clic sur Basculer en mode édition. Ceci se trouve dans le menu contextuel qui apparaît après un clic-droit sur la couche dans la légende. Sinon, vous pou-

Astuce 15 ÉDITIONS CONCURRENTES
Cette version de QGIS ne regarde pas si quelqu'un d'autre est en train d'éditer une entité en même temps que vous. La dernière personne à sauvegarder ses éditions gagne.

Astuce 16 ZOOMEZ AVANT D'ÉDITER
Avant d'éditer une couche, vous devez zoomer sur votre zone d'intérêt. Cela évite les temps d'attente lorsque les marqueurs de sommets s'affichent pour toute la couche.

vez utiliser le bouton Basculer en mode édition de la barre d'outils pour lancer ou stopper l'édition. Une fois la couche en mode édition, les marqueurs apparaissent sur les sommets et de nouveaux outils de la barre d'outil édition sont disponibles.

Zoomer avec la molette de la souris

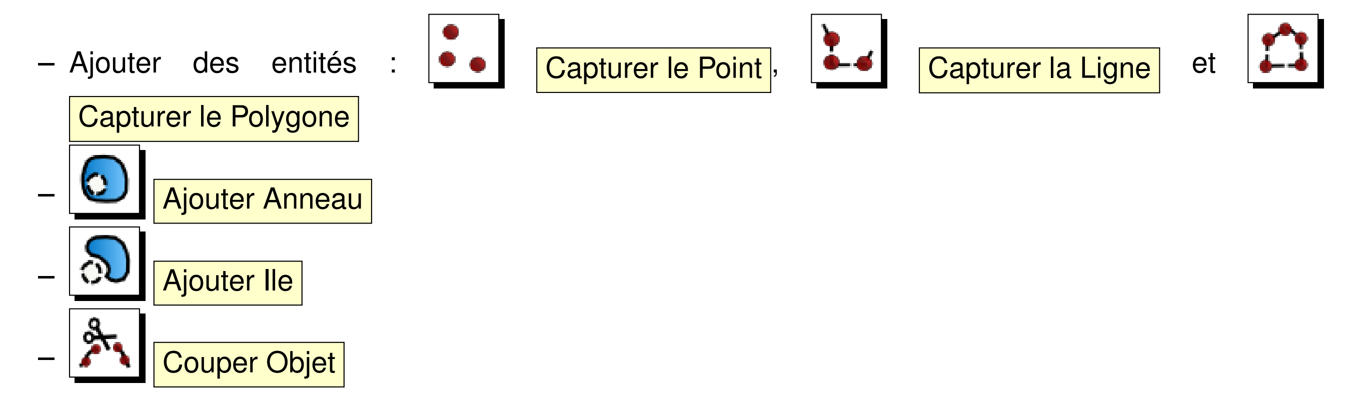
Lorsque vous numérisez, vous pouvez utiliser la molette de la souris pour zoomer et dézoomer sur la carte. Placez le curseur de la souris dans la zone de carte et faites rouler la molette vers l'avant (loin de vous) pour zoomer et vers l'arrière (vers vous) pour dézoomer. La position du curseur de la souris correspondra au centre du zoom. Vous pouvez paramétrer le comportement de la molette de la souris en allant dans l'onglet Outils cartographiques dans le menu Préférences > Options.

Se déplacer avec les flèches de direction

Se déplacer sur la carte pendant la numérisation est possible avec les flèches de direction. Placez le curseur de la souris sur la zone de carte et cliquez sur la touche flèche droite pour vous déplacer vers l'est, la flèche gauche pour aller vers l'ouest, la flèche haut pour le nord et la flèche bas pour le sud.

Vous pouvez également utiliser la barre d'espacement pour activer temporairement le déplacement sur la carte par les mouvements de la souris. Les touches PgUp et PgDown de votre clavier provoqueront le zoom ou le dé-zoom sans interrompre la session d'édition.

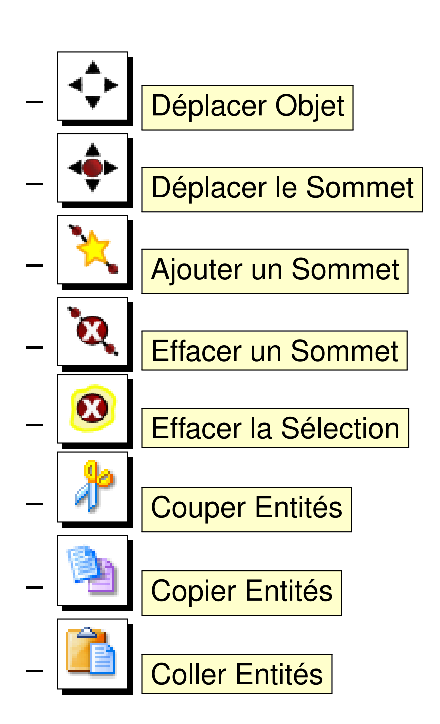
Vous pouvez réaliser les fonctions d'édition suivantes :



– Ajouter des entités : Capturer le Point, Capturer la Ligne et Capturer le Polygone
– Ajouter Anneau
– Ajouter Ile
– Couper Objet

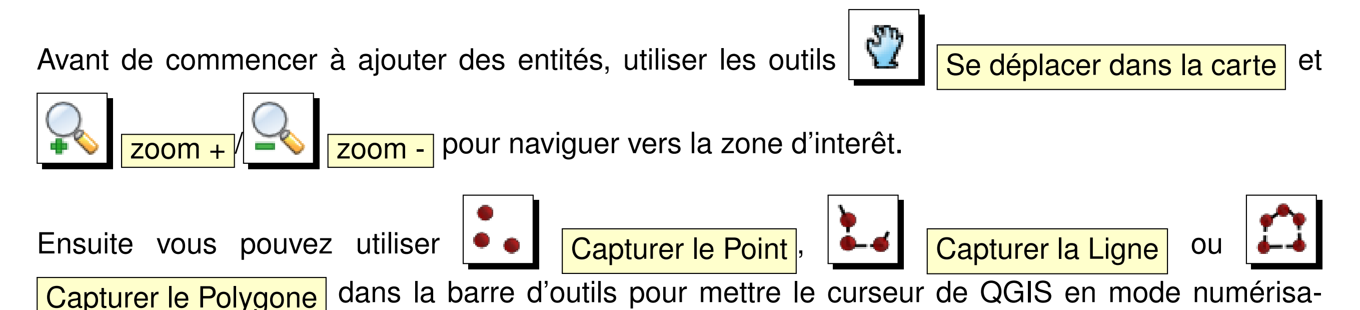
Astuce 17 MARQUEURS DE SOMMET

La version actuelle de QGIS gère deux type de marqueurs de sommet : un cercle semi-transparent ou une croix. Pour changer le style du marqueur, aller dans Options du menu Préférences et cliquez sur l'onglet Numérisation puis sélectionnez les paramètres appropriés.



- Déplacer Objet
- Déplacer le Sommet
- Ajouter un Sommet
- Effacer un Sommet
- Effacer la Sélection
- Couper Entités
- Copier Entités
- Coller Entités

Ajouter des entités



Avant de commencer à ajouter des entités, utiliser les outils Se déplacer dans la carte et zoom + / zoom - pour naviguer vers la zone d'interêt.

Ensuite vous pouvez utiliser Capturer le Point, Capturer la Ligne ou Capturer le Polygone dans la barre d'outils pour mettre le curseur de QGIS en mode numérisation.

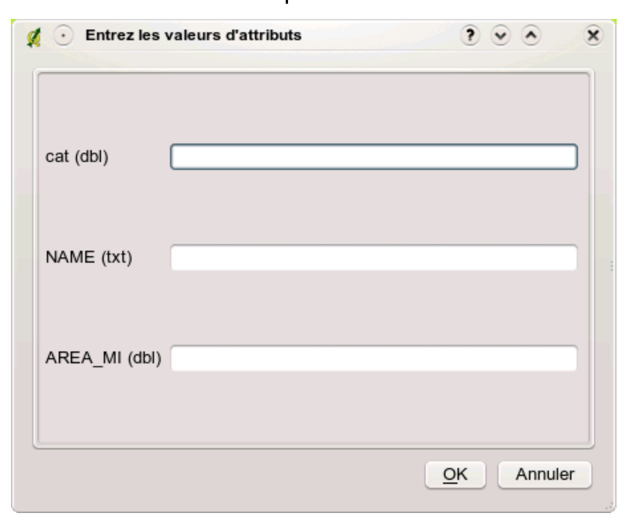
Pour chaque entité, vous numérisez d'abord la géométrie puis entrez les attributs.

Pour numériser la géométrie, faites un clic-gauche sur la zone de la carte pour créer le premier point de votre nouvelle entité.

Pour les lignes ou les polygones, continuer à faire des clics-gauche pour chaque nouveau point que vous souhaitez capturer. Lorsque vous avez fini d'ajouter des points, faites un clic-droit n'importe où sur la carte pour confirmer que vous avez fini d'entrer la géométrie de cette entité.

Les fenêtres des attributs apparaissent, ce qui vous permet d'entrer les informations sur la nouvelle entité. La figure 10 montre les attributs d'édition pour une nouvelle rivière fictive en Alaska.

FIG. 10: Fenêtre Entrez les valeurs d'attributs après la numérisation d'une nouvelle entité vecteur

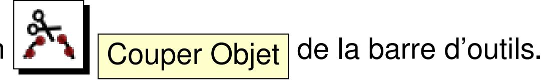


Astuce 18 TYPES DES VALEURS D'ATTRIBUT
Pour l'édition des shapefiles au moins, les types des attributs sont validés au moment de la saisie. À cause de cela, il n'est pas possible d'entrer un nombre dans un champ de type texte dans la fenêtre **Entrez les valeurs d'attributs** et vice-versa. Si vous avez besoin de le faire, vous devez éditer les attributs par la suite dans la fenêtre **Table d'attributs**.

Déplacer des objets

Vous pouvez déplacer des objets en utilisant le bouton  Déplacer Objet de la barre d'outils.

Couper des objets

Vous pouvez couper des objets en utilisant le bouton  Couper Objet de la barre d'outils.

Éditer les sommets d'un objet

Pour les couches PostgreSQL/PostGIS et shapefile, on peut éditer les sommets des entités.

Les sommets peuvent être édités directement, ce qui signifie que vous n'avez pas à choisir quelle entité vous voulez éditer avant que vous puissiez changer sa géométrie. Dans certains cas, plusieurs entités peuvent partager le même sommet et voilà les règles qui s'appliquent lorsqu'un bouton de la souris est pressé proche d'une entité :

– **Lignes** - La ligne la plus proche de la position de la souris est utilisée comme entité cible. Ensuite (pour déplacer ou supprimer un sommet) le sommet le plus proche sur cette ligne est la cible de l'édition.
– **Polygones** - Si la souris est à l'intérieur d'un polygone, celui-ci est l'entité ciblée ; autrement, le polygone le plus proche est utilisé. Ensuite (pour déplacer ou supprimer un sommet) le sommet le plus proche sur ce polygone est la cible de l'édition.

Vous aurez à définir le paramètre Préférences > Options > Numérisation > Rayon de recherche 10 à un nombre supérieur à zéro. Sinon QGIS ne sera pas en mesure de dire quelle entité est éditée.

Ajouter des sommets à un objet

Vous pouvez ajouter de nouveaux sommets en utilisant le bouton Ajouter un Sommet de la barre d'outils.

Notez qu'il n'y a aucun sens à ajouter des sommets à des entités de type ponctuelles !

Dans cette version de QGIS, les sommets peuvent uniquement être ajoutés à un segment de ligne *existant*. Si vous voulez étendre une ligne au-delà de ses extrémités, vous devez d'abord déplacer le sommet terminal puis ajouter un nouveau sommet là où le sommet terminal était.

Déplacer des sommets d'une entité

Vous pouvez déplacer des sommets en utilisant le bouton Déplacer le Sommet de la barre d'outils.

Effacer des sommets d'une entité

Vous pouvez supprimer des sommets en utilisant le bouton Effacer un Sommet de la barre d'outils.

Notez qu'il n'y a pas de sens à supprimer un sommet d'une entité ponctuelle ! Supprimer l'entité complète à la place.

De la même manière une ligne à un sommet ou un polygone à deux sommets n'a pas d'intérêt et entraînerait des comportements imprévisibles dans QGIS, donc ne faites pas ça.

Attention : Un sommet est identifié pour la suppression dès que vous cliquer à proximité d'une entité. Pour annuler cela, vous devez sortir du mode édition sans sauvegarder vos changements. (Bien entendu cela signifie que tous les changements non sauvegardés seront perdus).

Ajouter un anneau

Vous pouvez créez des polygones de type anneau en utilisant le bouton Ajouter Anneau de la barre d'outils. Ceci signifie qu'il est possible de numériser des polygones à l'intérieur d'une entité existante, qui seront alors des 'trous' de sorte que seule la zone entre les limites externes et internes du polygone reste, créant un polygone anneau.

Ajouter une île

Vous pouvez Ajouter une île à un multipolygone sélectionné. Le nouveau polygone île soit être numérisé en dehors du multipolygone sélectionné.

Couper, Copier et Coller des entités

Une entité sélectionnée peut être coupée, copiée et collée entre des couches d'un même projet QGIS, du moment que les couches de destination sont Basculées en mode édition au préalable.

Les entités peuvent également être collées dans des applications externes au format texte. Les entités sont alors représentées au format CSV et leur géométrie apparaît dans le format OGC Well-Known Text (WKT).

Cependant, dans cette version de QGIS, les entités au format texte venant d'applications externes ne peuvent pas être collées à une couche dans QGIS. En quoi les fonctions copier et coller sont utiles ? Et bien il se trouve que vous pouvez éditer plus d'une couche à la fois et que vous pouvez alors utiliser les fonctions copier/coller entre les couches. Pourquoi voudrions-nous faire cela ? Imaginons que nous devions travailler sur une nouvelle couche mais que nous avions besoin que d'un ou deux lacs, pas les 5 000 de notre couche `big_lakes`. Nous pouvons créer une nouvelle couche puis utiliser copier/coller pour y insérer les quelques lacs.

Voici un exemple de copie de quelques lacs dans une nouvelle couche :

1. Chargez la couche dont vous voulez copier des entités (couche source)
2. Chargez ou créez la couche sur laquelle vous voulez coller des entités (couche cible)
3. Lancez l'édition pour la couche cible
4. Assurez-vous que la couche source est active en cliquant dessus dans la légende
5. Utilisez l'outil Sélection pour sélectionner les entités dans la couche source
6. Cliquez sur l'outil Copier Entités
7. Assurez-vous que la couche cible est active en cliquant dessus dans la légende

8. Cliquez sur l'outil  Coller Entités
9. Stoppez l'édition et sauvegardez les changements

Qu'arrive-t-il si les couches source et cible ont différents schémas de données (noms et type des champs différents) ? QGIS remplit ceux qui correspondent et ignore les autres. Si la copie des attributs ne vous intéresse pas, la façon dont vous designer les champs et les types de données n'a pas d'importance. Si vous voulez être sûr que tout - entité et ses attributs - est copié, assurez-vous que les schémas de données correspondent.

Astuce 19 CONGRUENCE DES ENTITÉS COPIÉES

Si vos couches source et cible utilisent la même projection, les entités collées auront la même géométrie que dans la couche source. Cependant, si la couche cible n'a pas la même projection, QGIS ne peut garantir que les géométries seront identiques. Cela est simplement dû aux erreurs d'arrondissement faites lors de la conversion de projection.

Supprimer des entités sélectionnées

Si nous voulons supprimer un polygone en entier, nous pouvons le faire en sélectionnant d'abord le polygone en utilisant l'outil Sélectionne les données. Vous pouvez sélectionner plusieurs objets pour la suppression. Une fois le ou les objets sélectionnés, utilisez l'outil Effacer la Sélection pour les supprimer. Il n'y a pas de fonction annuler mais n'oubliez pas que votre couche n'est réellement changée tant que vous n'arrêtez pas l'édition et sauvegardez vos changements. Donc si vous faites une erreur, vous pouvez toujours annuler la sauvegarde.

L'outil Couper Entités de la barre d'outils numérisation peut également être utilisé pour supprimer des entités. Ceci supprime effectivement les entités et les place également dans un "pressepapier spatial". Donc nous coupons les entités pour les supprimer. Nous pouvons ensuite utiliser l'outil Coller Entités pour les récupérer, nous donnant alors la capacité d'annuler une fois les changements. Couper, copier et coller marchent sur les entités sélectionnées se qui signifie que nous pouvons travailler sur plus d'un objet à la fois.

Astuce 20 GESTION DE LA SUPPRESSION D'ENTITÉS

Lors de l'édition de shapefile, la suppression d'entités ne fonctionne que si QGIS est lié à une version 1.3.2 ou supérieure de GDAL. Les versions OS X et Windows de QGIS disponibles depuis le site de téléchargement incluent GDAL 1.3.2 ou supérieur.

Mode d'accrochage

QGIS permet aux sommets numérisés d'être accrochés aux autres sommets de la même couche. Pour définir la tolérance d'accrochage, aller dans Préférences > Options -> Numérisation . No-

tez que la tolérance d'accrochage est dans les unités de la carte.

Sauvegarder les couches éditées

Quand une couche est en mode édition, tous les changements sont stockés en mémoire par QGIS. Ils ne sont pas sauvegardés immédiatement dans la source de données ou sur le disque. Lorsque vous déactivez le mode édition (ou quittez QGIS), il vous est demandé si vous souhaitez sauvegarder les changements ou les annuler.

Si les changements ne peuvent pas être sauvés (par exemple à cause d'un disque plein ou des valeurs d'attributs dépassant la plage prévue), l'état de la mémoire de QGIS est préservé. Cela vous permet d'ajuster vos éditions et réessayer.

5.4.4 Créer une nouvelle couche

Pour créer une nouvelle couche à éditer, allez dans Nouvelle couche vectorielle du menu Couche. La fenêtre **Nouvelle couche vecteur** apparaitra telle que montrée dans la figure 11. Choisissez le type de couche (point, ligne ou polygone).

Notez que QGIS ne gère pas encore la création d'entité 2.5D (i.e. des entités avec des coordonnées X, Y, Z). Pour le moment, seuls des shapefiles peuvent être créés. Dans une version future de QGIS, la création de n'importe format de couches géré par OGR ou PostgreSQL sera possible.

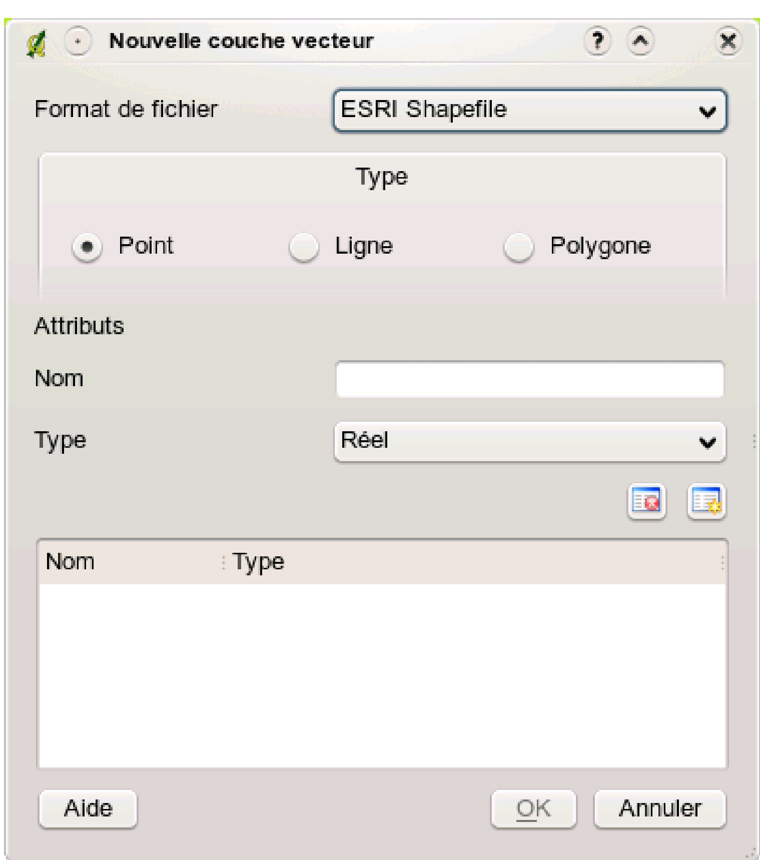
La création de couches GRASS est gérée par l'intermédiaire de l'extension GRASS. Référez-vous à la section 9.6 pour plus d'informations sur ce sujet.

Pour terminer la création de la nouvelle couche, ajouter les attributs désirés en cliquant sur le bouton Ajouter un attribut et en spécifiant le nom et le type de l'attribut. Seuls les attributs de type Type réel, Type entier, et Type string sont gérés. Une fois satisfait de vos attributs, cliquez sur OK et donnez un nom pour le shapefile. QGIS va automatiquement ajouter l'extension `.shp` au nom que vous lui avez spécifié. Une fois la couche créée, elle sera ajoutée à la carte et vous pouvez l'éditer de la manière décrite dans la Section 5.4.3 ci-dessus.

5.5 Constructeur de requêtes

Le constructeur de requêtes vous permet de définir un sous-ensemble de la table et l'afficher comme une couche dans QGIS. Il peut actuellement être utiliser avec les couches PostGIS. Par exemple, si vous avez une couche `towns` avec un champ `population`, vous pouvez sélectionner uniquement les plus grandes villes en entrant `population > 100000` dans le cadre SQL du constructeur de requête. La figure 12 montre un exemple de requête avec les données d'une couche PostGIS dont les attributs

FIG. 11: Fenêtre Nouvelle couche vecteur

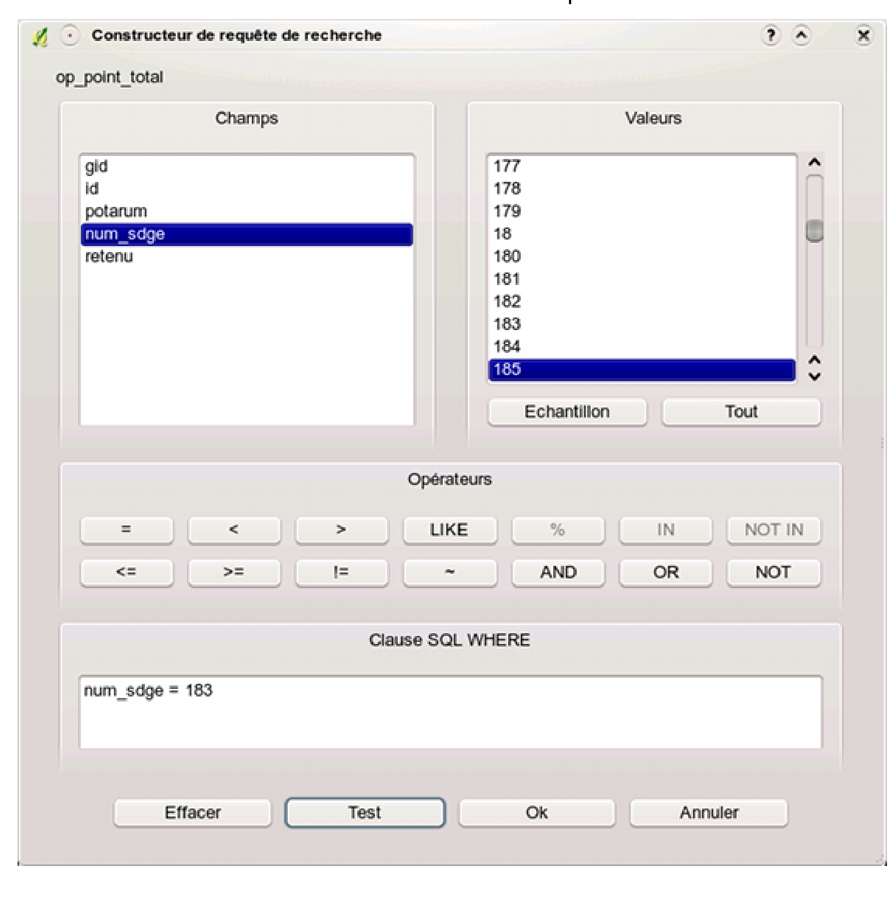


sont stockés dans PostgreSQL.

Le constructeur de requête liste les champs de la couche dans la base de données sur la gauche. Vous pouvez obtenir un extrait des données contenues dans le champ surligné en cliquant sur le bouton [Échantillon]. Cela récupère les 25 premières valeurs distinctes pour le champ dans la base de données. Pour avoir une liste de toutes les valeurs possible dans un champ, cliquez sur le bouton [Tout]. Pour ajouter un champ ou une valeur sélectionné à la requête, double-cliquez dessus. Vous pouvez utiliser différents boutons pour construire une requête ou vous pouvez simplement taper dans le cadre SQL.

Pour tester une requête, cliquez sur le bouton [Test]. Ceci va retourner un compte du nombre d'enregistrements qui seront sélectionnés dans la couche. Une fois satisfait de la requête, cliquez sur [OK]. Le code SQL pour la clause where apparaîtra dans la colonne SQL de la couche.

FIG. 12: Constructeur de requêtes



Astuce 21 CHANGER LA DÉFINITION D'UNE COUCHE

Vous pouvez changer la définition d'une couche après son chargement en modifiant la requête SQL utilisée pour définir la couche. Pour faire cela, ouvrez la fenêtre **Propriétés de la couche** en double-cliquant sur la couche dans la légende puis cliquez sur le bouton [Constructeur de requête] dans l'onglet [Général]. Voir Section 5.3 pour plus d'informations.

5.6 Sélection par requête

Dans QGIS, il est possible de sélectionner des entités en utilisant une interface similaire à celle du constructeur de requêtes utilisé dans 5.5. Dans la section ci-dessus, le but du constructeur de requêtes était seulement de montrer les entités répondant aux critères de filtre comme une 'couche virtuelle' / sous-ensemble. Le but de la fonction de sélection par requête est de surligner toutes les entités qui répondent à un critère particulier. La sélection par requête peut être utilisée sur tous les fournisseurs de données vecteur.

Pour faire une 'sélection par requête' sur une couche chargée, cliquez sur le bouton 

Ouvrir la table des attributs pour ouvrir la table de la couche. Ensuite cliquez sur le bouton Avancée... en bas de la fenêtre. Cela lance le Constructeur de requête qui permet de définir un sous-ensemble de la table et l'affiche comme décrit dans la Section 5.5.

6 Travailler sur des données raster

Cette section explique comment visualiser et définir les propriétés d'une couche raster. QGIS gère différents formats raster. Aujourd'hui les formats testés incluent :
- Arc/Info Binary Grid
- Arc/Info ASCII Grid
- Raster GRASS
- GeoTIFF
- JPEG
- Spatial Data Transfer Standard Grids (avec quelques limitations)
- DEM ASCII de l'USGS
- Erdas Imagine

Puisque l'implémentation du raster dans QGIS est basée sur la bibliothèque GDAL, les autres formats raster implémentés dans GDAL fonctionnent aussi probablement - dans le doute essayez d'ouvrir un fichier test et voyez s'il est géré. Vous trouverez plus d'information sur les formats gérés par GDAL en appendice A.2 ou sur `http://www.gdal.org/formats_list.html`. Si vous désirez charger des données raster GRASS, référez vous à la section 9.2.

6.1 Que sont les données raster ?

Les données raster dans les SIG sont des matrices de cellules discrètes qui représentent des objets, au dessus ou en dessous de la surface de la terre. Chaque cellule dans la grille raster est de la même taille et les cellules sont généralement rectangulaires (dans QGIS elles seront toujours rectangulaires). Un jeu de données raster typique incluent les données des capteurs distants telles que les photographies aériennes ou les images de satellites et les données modélisées telles que les matrices d'élévation.

Contrairement aux données vecteurs, les données raster n'ont typiquement pas de base de données d'enregistrement associées. Elles sont géocodées par leur résolution de pixel et leurs coordonnées x/y du coin du pixel de la couche raster. Cela permet à QGIS de positionner les données correctements dans la zone de la carte.

QGIS utilise les informations de géoréférencement dans les couches raster (par exemple GeoTiff) ou dans un fichier world approprié pour afficher correctement les données.

6.2 Charger des données raster dans QGIS

Les couches raster sont chargées soit en cliquant sur l'icône Charger une couche raster soit en sélectionnant l'option du menu Couches > Ajouter une couche raster. Plus

d'une couche peut être chargée en même temps en appuyant sur la touche Control ou Shift et en cliquant sur de plusieurs couches dans la boîte de dialogue **Ouvrez des sources de données taster gérés par GDAL**.

Une fois la couche raster chargée dans la légende de la carte vous pouvez cliquer sur le nom de la couche avec le bouton droit de la souris pour sélectionner et activer des paramètres spécifiques à la couche ou pour ouvrir une boîte de dialogue pour définir des propriétés du raster pour la couche.

Menu du bouton droit de la souris pour les couches raster

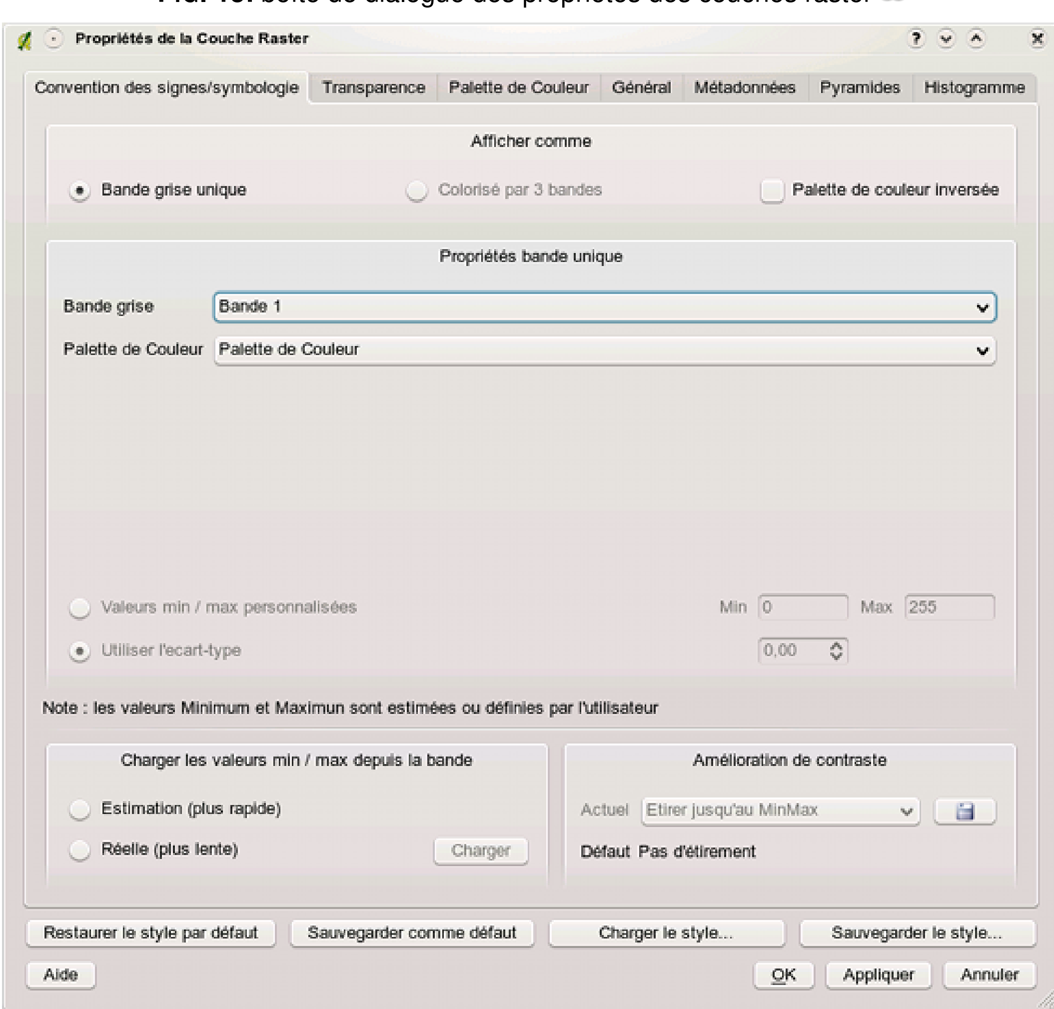
- Zoom sur l'étendue de la couche
- Zoom à la meilleur échelle (100%)
- L'affiche dans l'aperçu
- Supprime
- Propriétés
- Renomer
- Ajouter un groupe
- Tout étêndre
- Tout diminuer
- Afficher les groupes du fichier

6.3 boîte de dialogue de propriétés des Raster

Pour voir et définir les propriétés d'une couche raster, double-cliquez sur le nom de la couche dan la légende de la carte ou cliquez droit sur le nom de lacouche et choisissez Propriétés du menu contextuel : Figure 13 montre la boîte de dialogue **Propriétés de la couche raster**. Il y a plusieurs onglets dans cette fenêtre :

- Sémiologie
- Transparence
- Carte de couleur
- Général
- Méta-données
- Pyramides
- Histograme

FIG. 13: boîte de dialogue des propriétés des couches raster



6.3.1 Onlget sémiologie

QGIS peut afficher des couches raster de deux manières différentes :

- Bande simple - une bande de l'image sera affichée en nuance de gris ou en pseudocouleurs.
- Trois bandes de couleurs - trois bandes de l'image seront affichées, chaque bande représentant le composant rouge, vert ou bleu qui sera utilisé pour créer une image de couleur.

Pour les deux types de rendu vous pouvez inverser la sortie couleur en utilisant la case à cocher [X] Inverser la carte de couleur

Rendu des bandes simples

Ce choix vous permet deux possibilités : vous pouvez d'abord sélectionner quelle bande vous voulez utiliser pour le rendu (si le jeu de données a plus d'une bande).

La seconde option vous offre une sélection des tables de couleurs disponibles pour le rendu.

Les paramètres suivants sont disponibles à travers la liste déroulante carte de couleur Niveau de gris ▼, où niveau de gris est le paramètre par défaut.

Sont aussi disponible :
- Pseudo-couleur
- Pseudo-couleur psychédélique
- Couleurs indexées

Quand vous sélectionnez couleurs indexées Colormap ▼, l'onglet Couleurs indexées est disponible. Plus d'informations dans le chapitre 6.3.3.

QGIS peut restreindre les données affichées pour afficher seulement les cellules dont la valeur sont dans un nombre donné de déviations standards de la moyenne pour la couche. Cela est utile quand vous avez une ou deux cellules avec des valeurs anormalement hautes dans une grille raster qui ont un impact négatif sur le rendu du raster. Cette option est seulement disponible pour les images en pseudo-couleur.

Couleur à trois bandes

Cette sélection vous offre un large choix d'options pour modifier l'apparence de votre couche raster. Par exemple, vous pouvez passer les bandes de couleurs d'un ordre standard RVB à un autre.

L'échantillonage des couleurs est également disponible.

Astuce 22 VISUALISER UNE SEULE BANDE D'UN RASTER MULTIBANDE

Si vous désirez visualiser une seule bande (par exemple la bande rouge) d'une image multibande, vous pouvez penser que vous pourriez définir les bandes Vertes et Bleue à "Non définie". Mais ce n'est pas la manière correcte. Pour afficher la bande Rouge définissez le type d'image à nuance de gris, puis sélectionnez Rouge comme bande à utiliser pour le Gris.

6.3.2 Onglet transparence

QGIS a la possibilité d'afficher chaque raster à des niveaux de transparence différents. Utiliser la barre coulissante de transparence pour indiquer à quel niveau de transparence les couches sous-jacentes (s'il y en a) pourront être visible à travers cette couche raster. Cela est très utile, si vous désirez superposer plus d'une couche raster, par exemple une carte des reliefs ombrés superposé par une carte raster classifiée. Cela rendra la carte encore plus proche de la 3e dimension.

De plus, vous pouvez entrer une valeur raster qui pourra être traité comme *NODATA*

Un moyen encore plus flexible pour personnaliser la transparence est possible dans la section Options de transparence personnalisée. La transparence de chaque pixel peut être définie dans cet onglet.

Par exemple, nous voulons définir l'eau de notre fichier raster d'exemple `landcover.tif` à une transparence de 20 %. Les étapes suivantes sont nécessaire :

1. Chargez le fichier raster `landcover`
2. Ouvrez la boîte de dialogue **propriétées** en double-cliquant sur le nom du raster dans la légende ou avec un clic droit et en choisissant Propriétées du menu contextuel.
3. Sélectionnez l'onglet Transparence .
4. Cliquez sur le bouton Ajouter des valeurs manuellement. Une nouvelle ligne apparait dans la liste des pixels.
5. Entrez la valeur du raster (nous utilisons 0 ici) et ajustez la transparence à 20 %.
6. Pressez le bouton Appliquer et regardez la carte.

Vous pouvez répéter les étapes 4 et 5 pour ajuster d'autres valeurs avec une transparence personnalisée.

Comme vous pouvez le voir il est assez facile de définir une transparence personnalisée, mais cela peut prendre un peu de temp. Par conséquent vous pouvez utiliser le bouton Exporter dans un fichier pour sauver vos paramètres de transparence dans un fichier. Le bouton Importer à partir d'un fichier charge vos paramètres de transparence et les applique à la couche raster actuel.

6.3.3 Carte de couleur

L'onglet Colormap est seulement disponible quand vous avez sélectionné un rendu à une seule bande dans l'onglet Sémiologie (voir chapitre 6.3.1).

Trois manières de faire une interpolation de couleur sont disponibles :
– discrète ;
– liénaire ;
– exacte.
Le bouton Ajouter une entrée ajoute une couleur à la table de couleur individuelle. Double-cliquez sur la colonne valeur vous permet d'insérer une valeur spécifique. Double-cliquez sur la colonne

couleur ouvre une boîte de dialogue **Sélectionner une couleur** où vous pouvez sélectionner une couleur à appliquer sur cette valeur.

Alternativement, vous pouvez cliquer sur le bouton charger une carte de couleur à partir de bande qui tente de charger la table à partir d'une bande (si celle-ci en a une).

Le bloc Génèrer une nouvelle carte de couleurs vous permet de créer de nouvelles cartes de couleurs par catégorie. Vosu avez seulement besoin de sélectionner le nombre de classes 15 dont vous avez besoin et d'appuyez sur le bouton [Classifier]. Actuellement seule un mode de classification intervals égaux est géré.

6.3.4 Onglet général

L'onglet [Général] affiche des informations basiques sur le raster sélectionné, incluant la source de la couche et le nom affiché dans la légende (qui peut être modifié). Cet onglet montre aussi un aperçu de la couche, le symbol de la légende, et la palette.

De plus la visibilité en fonction de l'échelle peut être définie dans cet onglet. Vous devez activer la case à cocher et définir une échelle appropriée à laquelle vos données seront affichées dans la fenêtre de la carte.

Le système de référence spatial est également affiché ici comme chaîne PROJ.4. Cela peut être modifié en cliquant sur le bouton [Changer].

6.3.5 Onglet méta-données

L'onglet [Méta-données] affiche toute la richesse d'information sur la couche raster, dont les statistiques sur chaque bande dans la couche raster actuelle. Les statistiques sont recueillies sur l'idée de la 'nécessité de savoir', de sorte qu'il est possible qu'une couche n'ait pas de statistique collectée.

Cet onglet est principalement pour informations. Vous ne pouvez pas modifier les valeurs qui y sont affichées. Pour mettre à jour les statistiques vous devez aller dans l'onglet [Histogramme] et pressez le bouton [Rafraichir] en bas à droite, voir chapitre 6.3.7.

6.3.6 Onglet pyramides

Les couches raster à haute résolution peuvent ralentir la navigation dans QGIS. En créant des copies de plus basse résolution des données (pyramides), les performances peuvent être considérablement

améliorées puisque QGIS sélectionne la résolution la plus pertinente à utiliser en fonction du niveau de zoom.

Vous devez avoir accès en écriture dans le répertoire où les données originelles sont stockées pour construire les pyramides.
Plusieurs méthodes de reéchantillonage peuvent être utilisées pour calculer les pyramides :
– moyen
– plus proche voisin ;
Quand la case [X] construire les pyramides en interne si possible est cochée, QGIS tente de construire les pyramides en interne.

S'il vous plait notez que construire des pyramides peut altérer les fichiers données originaux et une fois créé ils ne peuvent plus être supprimé. Si vous désirez préserver une version 'sans pyramide' de vos raster, réalisez une copie de sauvegarde avant de les construire.

6.3.7 Onglet histograme

L'onglet Histogramme vous permet de visualiser la distribution des bandes ou des couleurs dans votre raster. Vous devez d'abord générer les statistiques du raster en cliquant le bouton Rafraichir . Vous pouvez choisir quelles bandes à afficher en les sélectionnant dans la liste déroulante en bas à gauche de l'onglet. Deux types de graphiques différents sont permis :

– graphique en barre ;
– graphique linéaire ;

Vous pouvez définir le nombre de colonnes du graphique à utiliser et décider si vous voulez [X] Permettre l'approximation ou afficher les valeurs [X] En dehors du domaine . Une fois que vous avez vu l'histogramme, vous remarquerez que les statistiques des bandes ont été remplies dans l'onglet méta-données .

Astuce 23 REGROUPEMENT DES STATISTIQUES RASTER
Rassembler des statistiques pour une couche, sélectionnez un rendu en pseudo-couleur et cliquez sur le bouton Appliquer . Regrouper des statistiques pour une couche peut prendre du temp. Soyez patient pendant que QGIS examine vos données !

7 Travailler avec des données OGC

QGIS gère le WMS et le WFS comme source de données. Leur gestion est native, mais le service WFS est géré sous forme d'extension.

7.1 Qu'est ce que les données OGC

more than 300 members processing Le Consortium Geospatial Ouvert (Open Geospatial Consortium - OGC) est une organisation internationale à laquelle participent plus de 300 organisations commerciale, gouvernementale, associative et laboratoire de recherche à travers le monde. Ses membres développent et implémentent des standards pour les services et le contenu géospatial, traitement de données SIG et de format d'échange.

specifications geospatial technology, `http://www.opengeospatial.org/`. Un nombre croissant de spécifications décrivant une modélisation de donnée basique pour les objets géographiques ont été développées pour servir des besoins spécifiques dans des situations nécessitant une interopérabilité et des technologies géospatiales, dont les SIG. Des informations supplémentaires peuvent être trouvées sur le site `http://www.opengeospatial.org/`.

Les spécifications importantes de l'OGC sont :

- **WMS** - Web Map Service
- **WFS** - Web Feature Service
- **WCS** - Web Coverage Service
- **CAT** - Web Catalog Service
- **SFS** - Simple Features for SQL
- **GML** - Geography Markup Language

PostGIS Les services OGC sont de plus en plus utilisés pour échanger des données géospatiales entre différentes implémentations SIG et des fournisseurs de données. QGIS peut maintenant traiter trois des spécifications citées, ceux-ci étant SFS (par la gestion du fournisseur de données PostgreSQL / PostGIS, voir section 5.2) ; comme client WFS et WMS.

7.2 Client WMS

7.2.1 Aperçu de la gestion WMS

QGIS peut actuellement agir comme client WMS et comprend les versions 1.1, 1.1.1 et 1.3 des serveurs WMS. Il a été particulièrement testé avec des serveurs accessibles publiquement comme ceux de DEMIS et JPL OnEarth.

Les serveurs WMS agissent en fonction des requêtes envoyées par le client (par exemple QGIS) pour une carte raster avec une étendue donnée, un ensemble de couches, une sémiologie et une transparence. Le serveur WMS consulte alors ses sources de données locales, rasterise la carte et la renvoie au client dans un format raster. Pour QGIS cela sera par exemple du JPEG ou du PNG.

Un WMS est de manière générale un service web mis en oeuvre selon une architecture REST (Representational State Transfer) plutôt qu'un service RPC (Remote Procedure Call) pleinement déployé

Des couches WMS peuvent être ajoutées assez simplement, du moment que vous connaissez l'URL pour accéder au serveur WMS, vous avez une connexion sous forme de service sur ce serveur, et celui-ci comprend le protocole HTTP comme mécanisme de transport.

7.2.2 Sélectionner des serveurs WMS

La première fois que vous utilisez la fonctionnalité des services WMS, il y a aucun serveur définie. Vous pouvez commencer en cliquant sur le bouton Ajoutez une couche WMS dans la barre des outils ou à travers le menu Couche > Ajoutez une couche WMS...

La boîte de dialogue **Ajoute une couche d'un serveur** pour ajouter des couches d'un serveur WMS s'ouvre. Heureusement, vous pouvez ajouter des serveurs pour jouer en cliquant le bouton Ajouter les serveurs par défaut . Cela ajoutera au moins trois serveurs WMS pour tester, incluant celui de la NASA (JPL). Pour définir un nouveau serveur WMS dans la section Connexions au serveur , sélectionnez Nouveau . Puis entrez les paramètres de connection de votre serveur WMS désiré, comme listé dans le tableau 2 :

TAB. 2: Paramètres de connection WMS

Nom	Un nom pour cette connexion. Ce nom sera utilisé dans la liste déroulante des connexions aux serveurs afin que vous puissiez distinguer des autres serveurs WMS.
URL	URL du serveur fournissant les données. Cela doit être un nom d'hôte publique ; de même format que vous utilisez pour ouvrir une connexion Telnet ou pinguer un hôte (ou dans un navigateur Internet).

Si vous devez configurer un serveur proxy pour pouvoir recevoir des services WMS à partir d'Internet, vous pouvez ajouter votre serveur proxy dans les options. Choisissez le menu Préférences > Options... et cliquez sur l'onglet Proxy . Vous pouvez alors ajouter votre configuration du proxy et l'activer en cochant la case [X] Utiliser un proxy pour l'accès Internet.

Une fois que la nouvelle connexion du serveur WMS a été créée, elle sera sauvegardée pour les futures sessions de QGIS.

Astuce 24 À PROPOS DES URL DES SERVEURS WMS
Assurez vous, lorsque vous entrez l'url du serveur WMS, d'avoir le début de l'URL. Par exemple, vous ne devez pas avoir ce genre de paramètre `request=GetCapabilities` ou `version=1.0.0` dans votre URL

Le tableau 3 montre quelques exemples d'url WMS pour démarrer. Ces liens ont été vérifié en décembre 2006, mais peuvent avoir changer :

TAB. 3: Exemple d'URL WMS publique

Nom	**URL**
Atlas du Canada	http ://atlas.gc.ca/cgi-bin/atlaswms_en ?
DEMIS	http ://www2.demis.nl/wms/wms.asp ?wms=WorldMap&
Geoscience Australia	http ://www.ga.gov.au/bin/getmap.pl ?dataset=national
NASA JPL OnEarth	http ://wms.jpl.nasa.gov/wms.cgi ?
Utilsiateurs QGIS	http ://qgis.org/cgi-bin/mapserv ?map=/var/www/maps/main.map&

Une liste de serveurs WMS exhaustive peut être trouvé ici `http://wms-sites.com`.

7.2.3 Charger des couches WMS

Une fois que vous avez remplie correctement vos paramètres vous pouvez sélectionner le bouton [Se connecter] pour récupèrer les possibilités du serveur sélectionné. Cela inclut le format d'image, les couches, les styles des couches et les projections. Puisque c'est une opération sur un réseau, la vitesse de la réponse dépend de la qualité de votre connection réseau au serveur WMS. Pendant le téléchargement des données du serveur WMS, la progression du téléchargement est visualisé en bas à gauche de la boîte de dialogue du plugin WMS.

Votre écran doit ressembler un peu plus à la figure 14, qui affiche la réponse fournit par le serveur WMS de la NASA JPL OnEarth.

Format d'image

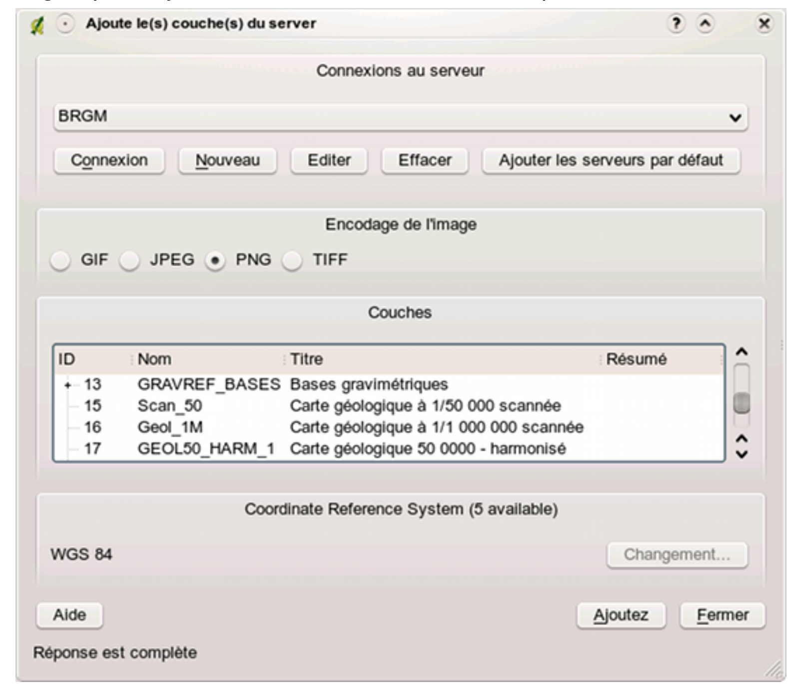
La section [Format d'image] liste maintenant les formats qui sont gérés à la fois par le client et leur serveur.Choisissez en un en fonction de votre préférence quant à la précision de l'image.

Couches

La section [couche] liste les couches disponibles dans le serveur WMS. Vous pouvez remarquer que certaines couches sont extensibles, cela signifie que la couche peut être affiché en fonction de plusieurs styles.

Vous pouvez sélectionner plusieurs couches à la fois, mais seulement un style d'image par couche.

FIG. 14: Dialogue pour ajouter un serveur WMS, en indiquant ses couches disponibles



Astuce 25 FORMAT D'IMAGE

Les serveurs WMS vous offriront typiquement le choix entre les formats d'image JPEG et PNG. JPEG est un format de compression avec perte, tandis que le format PNG reproduit pleinenement les données raster brutes.
Utilisez le format JPEG si vous pensez que la données WMS est une orthophotographie ou qu'une perte de qualité d'image ne vous pose pas de problème. Ce compromis vous permet de réduire par 5 le taux de transfert nécessaire comparé au format PNG.
Utilisez le format PNG si vous désirez une représentation précises des données originales et que l'augmentation du taux de transfert ne vous pose pas de problème.

Lorsque plusieurs couches sont sélectionnées, celles-ci seront combinées par le serveur WMS et transmises à QGIS en une seule fois.

Astuce 26 ORDONNER LES COUCHES WMS

Dans cette version de QGIS, les couches WMS créées par un serveur sont superposées dans l'ordre listé dans la section couches, du haut vers le bas de la liste. Si vous voulez superposer des couches dans l'ordre inverse, vous pouvez alors sélectionner Ajouter une couche WMS une seconde fois, choisir le même serveur, et sélectionner un groupe de couches que vous voulez au-dessus du premier groupe.

Transparence

Dans cette version de QGIS, le paramètre de transparence est codé en dur pour être toujours activé, si disponible.

Astuce 27 TRANSPARENCE DES COUCHES WMS

la disponibilité de la transparence de l'image WMS dépend du format d'image utilisé : les formats PNG et GIF gère la transparence, tandis que le format JPEG ne le gère pas.

Système de Référence de Coordonnées

Un système de Référence de Coordonnées (CRS) est la terminologie de l'OGC pour une projection QGIS.

Chaque couche WMS peut être représenté dans plusieurs projections (ou CRS), en fonction de la possibilité du serveur WMS. Vous pouvez avoir noté que les *x* changent dans l'en-tête du *Système de Référence des Coordonnées(x disponible)* lorsque vous sélectionnez et déselectionnez les couches de la section couches .

Pour choisir une projection, sélectionnez Changer... et un écran similaire à la figure 17 dans la section 8.3 apparaitra. La principale différence avec l'écran de la version WMS est que seules les projections gérées par le serveur sera listées.

Astuce 28 LES PROJECTIONS WMS

Pour de meilleurs résultats, faîtes en sorte que la couche WMS soit la première couche que vous ajoutez dans le projet. Cela permet à la projection du projet d'hériter la définition de la projection que vous avez utilisé pour le rendu de la couche WMS. La projection à la volée (voir Section 8.2) peut être utilisée pour placer les couches vectorielles supplémentaires dans la projection du projet. Dans cette version de QGIS, si vous ajoutez une couche WMS plus tard et lui donner une projection différente de celui du projet en cours, cel apeut entraîner des résultats aléatoires.

7.2.4 Utiliser l'outil Identifier

Une fois que vous avez ajouté un serveur WMS et si une couche du serveur WMS est intérogeable, vous pouvez utiliser l'outil Identifier pour sélectionner un pixel sur la carte. Une requête est envoyée au serveur WMS pour chaque sélection effectuée.

Les résultats de la requête sont renvoyés au format texte. Le formattage de ce texte est dépendant du serveur WMS utilisé.

7.2.5 Visualiser les propriétés

Une fois que vous avez ajouté un serveur WMS, vous pouvez voir ses propriétés en cliquant avec le bouton droit de la souris sur sa légende, et en sélectionnant le bouton Propriétés .

Onglet métadonnées

L'onglet métadonnées affiche la richesse des informations du serveur WMS, généralement collecté à partir de la requête Capabilities renvoyée par le serveur.

Beaucoup de définitions peuvent être obtenues par la lecture des normes WMS **?**, **?**, mais en voici quelques-unes :

- **Propriétés du serveur**
  - **Version du WMS** - la version du serveur WMS géré par le serveur.
  - **Formats d'image** - la liste des types MIME que le serveur peut renvoyer lors qu'il dessinne la carte. QGIS gère tous les formats pour lesquelles la bibliothèque Qt en sous-couche a été compilée, qui est typiquement à minima les types `image/png` et `image/jpeg`.
  - **Formats de l'outil Identitier** - la liste des types MIME auxquels le serveur peut répondre quand vous utilisez l'outil Identifier. Pour l'instant QGIS gère le type `text-plain`.
- **Propriétés de la couche**
  - **Selectioné** - si cette couche a été sélectionnée ou pas quand son serveur a été ajouté au projet.
  - **Visible** - si cette couche a été sélectionnée ou non comme visible dans la légende (pas encore utilisé dans cette version de QGIS).
  - **Peut identifier** - si cette couche retournera ou pas des résultats quand l'outil Identifier est utilisé sur celle-ci.
    is
  - **Peut être transparent** - si cette couche peut être rendu ou non avec une transparence. Cette version de QGIS utilisera toujours la transparence si cette option est à *Oui* et que le format d'image gère la transparence .
  - **Peut zoomer** - si on peut zoomer ou non sur cette couche avec le serveur. Cette version de QGIS assume que toutes les couches WMS ont ce paramètre définie à *Oui*. Les couches déficientes peuvent être rendu d'une manière étrange.
  - **décompte des cascades** - les serveurs WMS peuvent agir comme un proxy à d'autres serveurs WMS pour obtenir des données pour une couche. Cette entrée affiche combien de fois la requête pour cette couche est redirigée vers un autre serveur WMS pour un résultat.
  - **Largeur fixe**, **hauteur fixe** - si une couche a des dimensions du pixel source limitées. Cette version de QGIs suppose que toutes les couches WMS ont ce paramètre fixé. Les couches déficientes peuvent être rendu d'une manière étrange.
  - **Limite du contour en WGS 84** - la limite du contour de la couche, en coordonnées WGS 84. Certains serveurs WMS ne définissent pas ceci correctement (par exemple des coordonnées UTM sont utilisées à la place). Si cela est le cas, alors la vue initiale sera rendu avec une vue

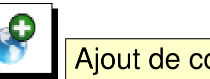
très étendue. Le webmaster du WMS doit être informé de cette erreur, qui sont connu en tant qu'éléments XML `LatLonBoundingBox`, `EX_GeographicBoundingBox` ou `BoundingBox` CRS :84 du WMS.

- **Disponibilité des CRS** - les projections que l'on peut utiliser par le serveur WMS. Ceux-ci sont listés dans le format natif du WMS.
- **Disponibilité des styles** - les styles des images de cette couche qui peuvent être utilisé pour le rendu par le serveur WMS.

7.2.6 Limitations du client WMS

Seule quelques fonctionnalités du client WMS ont été incluses dans cette version de QGIS. Les exceptions les plus notables sont présentées ci-après :

Éditer la configuration d'une couche WMS

Une fois que vous avez complété la procédure d'  Ajout de couches WMS, il n'y aucun moyen de modifier la configuration.

Une solution de contournement est de supprimer la couche et de recommencer.

Serveurs WMS nécessitant une authentification

Seuls les serveurs WMS public sont accessibles. Il n'y a pas de possibilité pour appliquer une combinaison nom d'utilisateur et mot de passe comme authentification à un serveur WMS.

Astuce 29 ACCÉDER DES COUCHES OGC SÉCURISÉES

Si vous avez besoin d'accéder à des couches sécurisées, vous pouvez utiliser InteProxy comme proxy transparent, qui gère plusieurs méthodes d'authentification. Vous pouvez trouver plus d'informations dans le manuel InteProxy que vous pouvez trouver sur le site `http://inteproxy.wald.intevation.org`.

7.3 Client WFS

Dans QGIS, une couche WFS se comporte à peu près comme n'importe quelle autre couche vecteur. Vous pouvez identifier et sélectionner des objets et voir la table attributaire. Une exception est que l'édition n'est pas prise en charge pour l'instant. Pour lancer le plugin WFS, vous avez besoin d'ouvrir des Plugins > Gestionnaire de plugin ..., activez le [x] plugin WFS et cliquez sur le bouton OK .

Une nouvelle icône  Ajouter une couche WFS apparaît à côté de celle du WMS. Cliquez des-

sus pour ouvrir la boîte de dialogue. En général ajouter une couche WFS est identique à la procédure utilisée pour une couche WMS. La différence est qu'il n'y a pas de serveurs par défaut, vous devez donc en ajouter.

7.3.1 Charger une couche WFS

Comme exemple, nous utilisons le serveur WFS de DM Solutions et affichons une couche. L'url est :

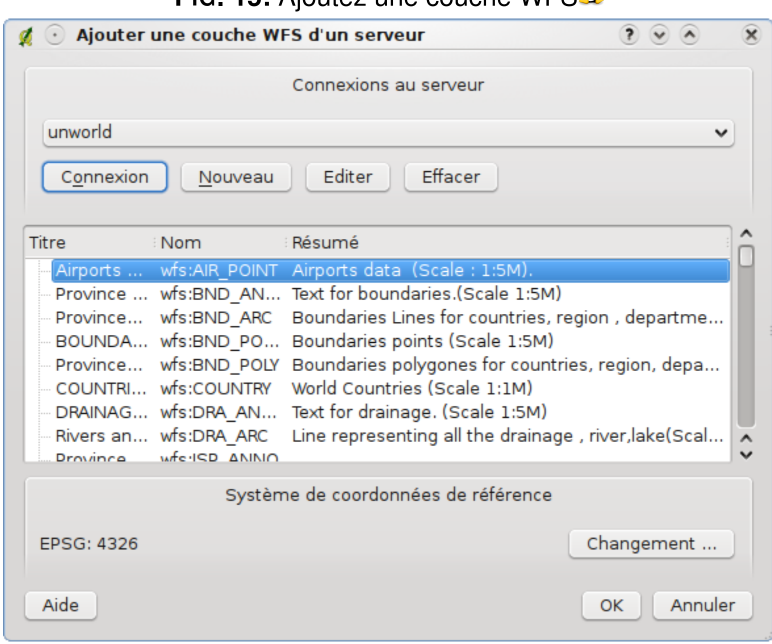
```
http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap?VERSION=1.0.0&SERVICE=
wfs&REQUEST=GetCapabilities
```

1. Assurez vous que le plugin WFS est activée ; Si ce n'est pas le cas, ouvrez le gestionnaire de plugin et activez le.
2. Cliquez sur le bouton Ajoutez une couche WFS dans la boîte à outil des plugins.
3. Cliquez sur Nouveau
4. Entrez Nom DM Solutions comme nom.
5. Entrez l'url (voir page précédente).
6. Cliquez sur OK
7. Choisissez Connections au serveur DM Solutions dans la liste déroulante.
8. Cliquez sur Connect .
9. Attendez que la liste des couches soit complètes.
10. Cliquez sur la couche Canadian Land .
11. Cliquez sur Ajoutez pour ajouter la couche à la carte.
12. Patientez que les objets apparaissent.

Vous remarquerez que la progression du téléchargement est affichée en bas à gauche de la fenêtre principale de QGIS.

Souvenez vous que les plugins fonctionnent mieux avec des serveurs WFS sous MapServer. Il est encore possible que vous ayez à faire face à quelques problèmes et crashes. Vous pouvez vous attendre à des améliorations dans les versions futures du plugin.

FIG. 15: Ajoutez une couche WFS



Astuce 30 TROUVER DES SERVEURS WMS ET WFS

Vous pouvez trouver des serveurs WMS et WFS supplémentaire en utilisant Google ou votre moteur de recherche préféré. Il y a un certain nombre de site qui liste des url publiques, certaines maintenues d'autres non.

8 Utiliser les projections

QGIS vous permet de définir une projection (CRS ou Système de coordonnées de référence) globale et pour des couches qui n'ont pas de projection prédéfinie au niveau du projet. Il permet également à l'utilisateur de définir des systèmes de coordonnées de référence personnalisés et gère la reprojection à la volée de couches vectorielles. Toutes ces fonctionnalités permettent à l'utilisateur d'afficher des couches avec différentes projections et de les superposer correctement.

8.1 Aperçu de la gestion des projections

QGIS gère approximativement 2 700 projections connues. Les définitions pour chacune d'entre elles sont stockées dans une base de données SQLite qui est installée avec QGIS. Normalement vous n'avez pas besoin de manipuler cette base de données directement. En fait, cela peut poser des problèmes de gestion de projections. Les projections personnalisées y sont stockées dans une base de données utilisateur. Voir la section 8.4 pour avoir des informations sur la gestion de vos systèmes de coordonnées de référence personnalisées.

Les projections disponibles dans QGIS sont basées sur celle définie par EPSG et sont en grande partie extraite de la table spatial_references dans PostGIS version 1.x. Les identifiants EPSG sont présents dans la base de données et peuvent être utilisés pour définir une projection dans QGIS.

Dans le but d'utiliser des projections à la volée, vos données doivent contenir des informations sur leurs systèmes de coordonnées de référence ou vous devez définir une projection pour votre projet, couche ou globale. Pour les couches PostGIS, QGIS utilise l'identifiant de référence spatiale qui a été définie quand la couche a été créée. Pour les données gérées par OGR, QGIS utilise un moyen spécifique au format de définition de la projection. Dans le cas de shapefile, cela signifie un fichier contenant une spécification Well Known Text (WKT) de la projection. Le fichier de projection a le même nom que le fichier shape et une extension prj. Par exemple, un shapefile nommé `alaska.shp` aura un fichier de projection correspondant nommé `alaska.prj`.

8.2 Définir une projection

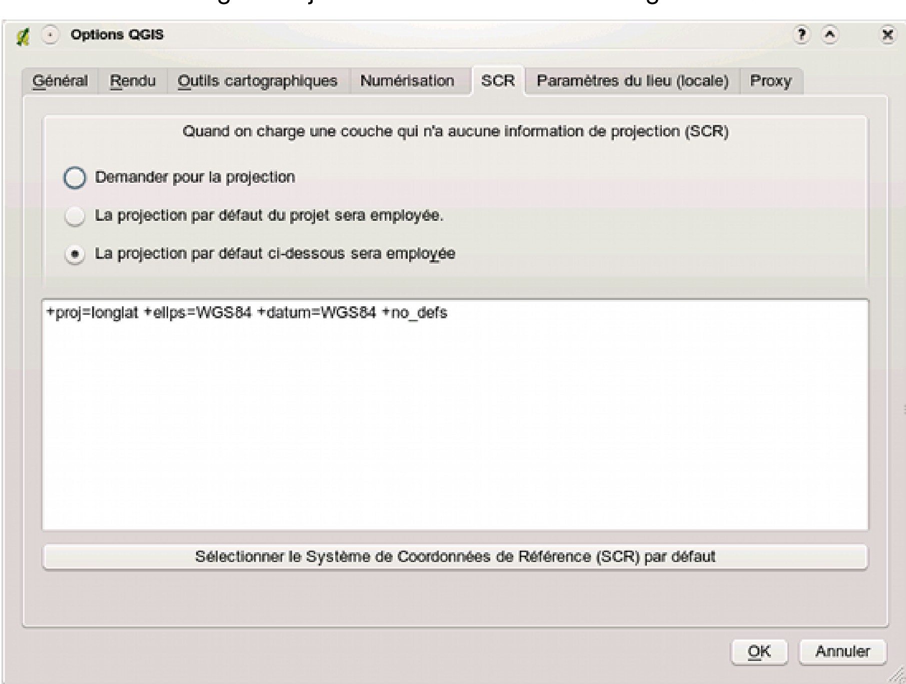
QGIS ne définit plus la projection de la carte au système de coordonnées de référence de la première couche chargée. Lorsque vous démarrez une session QGIS avec des couches qui ne possèdent pas de projection, vous devez contrôler et définir la définition de la projection pour ces couches. Cela peut être réalisée globalement ou par projet dans l'onglet projection dans Préférences > Options (voir Figure 16).

- [X] Demande de projection
- [X] La valeur par défaut au niveau du projet sera utilisée

– [X] La valeur globale par défaut affiché ci-dessous sera utilisée

La projection globale par défaut `proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs` est prédéfinie dans QGIS mais peut bien sur être changée, et la nouvelle définition sera sauvée pour les prochaines sessions de QGIS.

FIG. 16: Onglet Projection dans la boîte de dialogue de QGIS



Si vous voulez définir le système de coordonnées de référence pour certaines couches sans information sur la projection, vous pouvez également faire cela dans l'onglet Général de la boîte de dialogue propriété de la couche raster (6.3.4) ou de celle de la couche vecteur (5.3.1). Si votre couche a déjà une projection définie, elle sera affichée comme montré dans la figure 6.

8.3 Définir une projection à la volée (OTF)

QGIS n'active pas la projection à la volée par défaut, et cette fonction est seulement gérée pour les couches vectorielles. Pour utiliser la projection à la volée, vous devez ouvrir la boîte de dialogue Propriétés du projet, sélectionner une projection et cocher la case à cocher [X] Activer la projection à la volée. Il y a deux manières pour ouvrir la boîte de dialogue :

1. Sélectionnez Propriétés du projet à partir du menu Préférences.

2. Cliquez sur l'icône projection dans le coin en base à droite de la barre de statut.

Si vous avez déjà chargée une couche, et désirez activer la projection à la volée, la meilleure façon de faire est d'ouvrir l'onglet Système de coordonnées de référence de la boîte de dialogue **Propriétés du projet**, sélectionner la projection de la couche chargée, et d'activer la case [X] Activer la projection à la volée. L'icône projection affichera un symbole vert et toutes les couches vecteurs chargées plus tard seront projetées à la volée dans la projection définie.

L'onglet Système de Coordonnées de Référence de la boîte de dialogue **Propriétés du projet** contient quatre composants importants comme numéroté à la figure 17 et décrit ci-dessous.

1. **Activer la projection à la volée** - cette case à cocher est utilisée pour activer ou désactiver la projection à la volée. Lorsqu'elle ne l'est pas, chaque couche est dessinée en utilisant les coordonnées lues dans la source de données. Lorsqu'elle est activée, les coordonnées de chaque couche sont projetées dans le système de coordonnées de référence définie pour la carte.
2. **Système de Coordonnées de Référence** - c'est une liste de toutes les projections gérées par QGIS, incluant les systèmes de coordonnées de référence géographiques, projetées et personnalisées. Pour utiliser une projection, sélectionnez-la dans la liste en déroulant le noeud approprié et en sélectionnant la projection. La projection active est présélectionnée.
3. **Texte Proj4** - c'est une chaîne de projection utilisé par le moteur de projection Proj4. Ce texte est en lecture seul et est fournit pour information.
4. **Rechercher** - si vous connaissez le code EPSG ou le nom d'un système de coordonnées de référence, vous pouvez utiliser la fonction rechercher pour le retrouver. Entrez le code et cliquez sur le bouton Trouver .

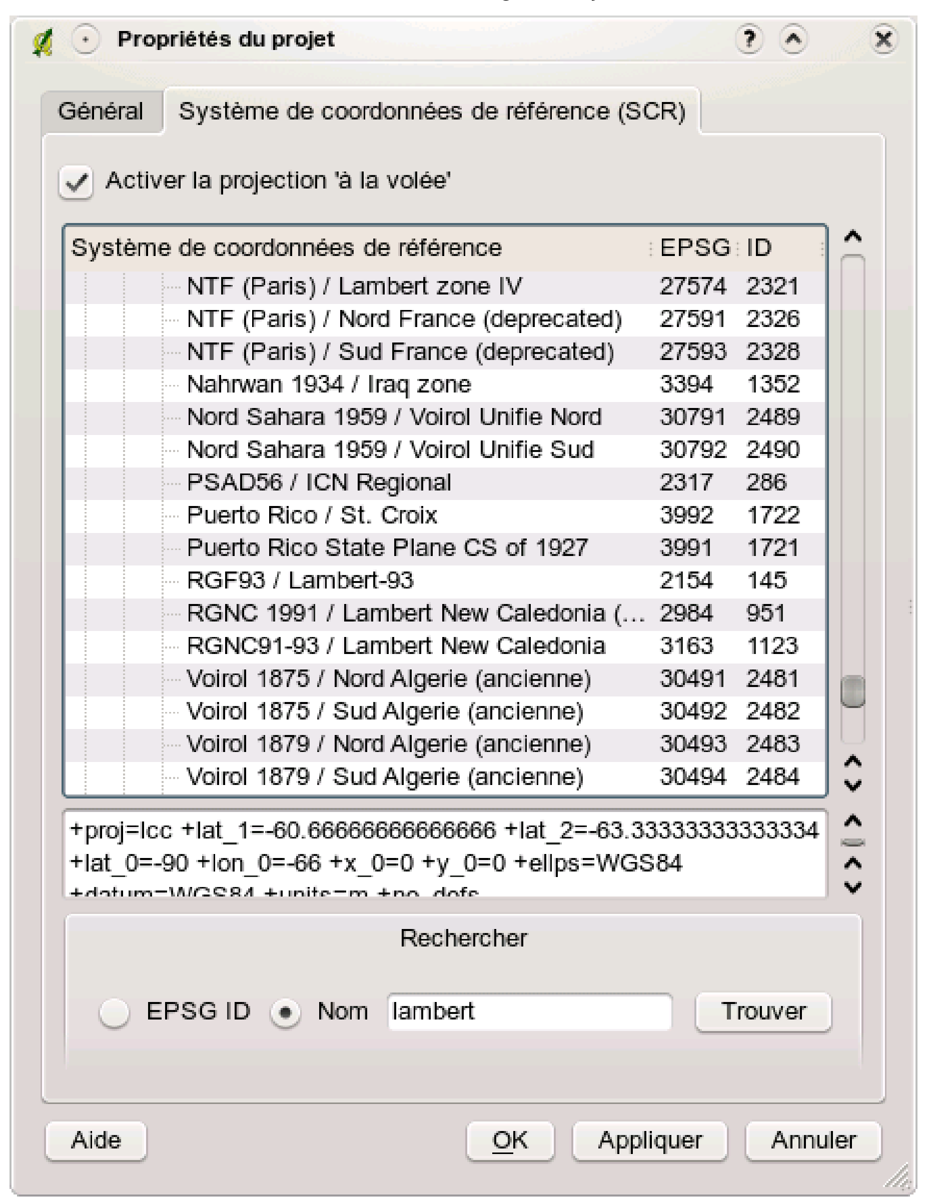
Astuce 31 BOÎTE DE DIALOGUE PROPRIÉTÉ DU PROJET

Si vous ouvrez la boîte de dialogue **Propriétés du projet** à partir du menu Préférences, vous devez cliquer sur l'onglet Système de Coordonnées de Référence pour voir les définitions de projection. Ouvrir la boîte de dialogue à partir de l'icône projection vous amenera directement dans l'onglet Système de Coordonnées de Référence .

8.4 Système de Coordonnées de Référence personnalisées

Si QGIS ne fournit pas le système de coordonnées de référence dont vous avez besoin, vous pouvez en définir un. Pour cela, sélectionnez ☆ Projection personnalisée à partir du menu Préférences.

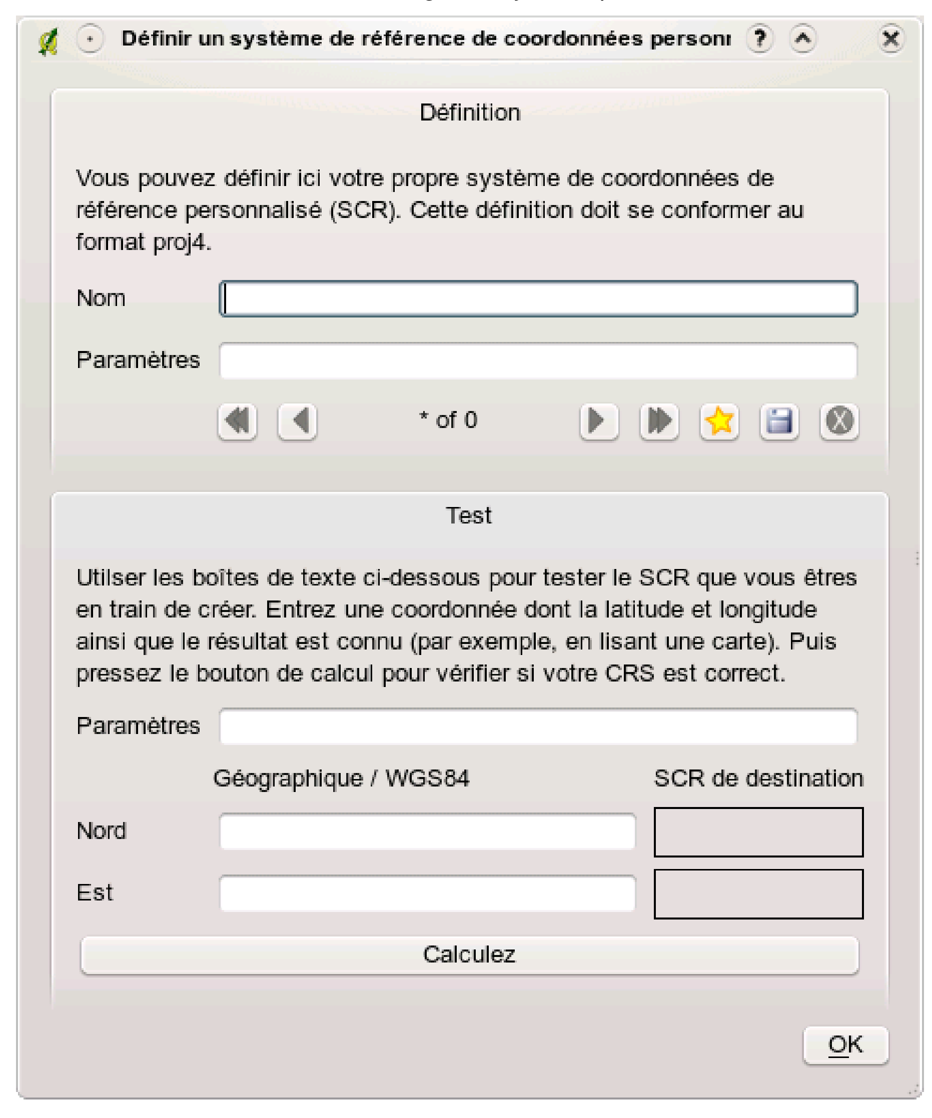
FIG. 17: Boîte de dialogue Projection



Les projections personnalisées sont stockées dans la base de données utilisateur de QGIS. En plus de votre projection personnalisée, cette base de données contient également vos signets spatiaux et autres données personnalisées.

Définir une projection personnalisée dans QGIS nécessite une bonne compréhension de la bibliothèque de projection Proj4. Pour commencer, référez vous aux Procédures de Projection Cartographique pour l'environnement UNIX - Un manuel d'utilisateur de Gerald I. Evenden, U.S. Geological Survey Open-File Report 90-284, 1990 (disponible sur `ftp://ftp.remotesensing.org/proj/`

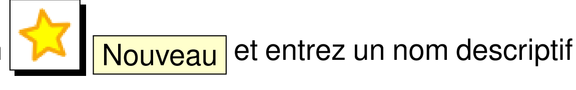
FIG. 18: Boîte de dialogue Projection personnalitée



`OF90-284.pdf`). Ce manuel décrit l'utilisation de `proj.4` et les applications en lignes de commandes liées. Les paramètres cartographiques utilisés avec `proj.4` sont décrit dans le manuel utilisateur et sont les mêmes que ceux utilisés par QGIS.

La boîte de dialogue **Définition d'un système de coordonnées de référence personnalisée** nécessite seulement deux paramètres pour définir une projection personnalisée :

1. un nom descriptif et
2. les paramètres cartographiques au format PROJ.4.

Pour créer une nouvelle projection, cliquez sur le bouton  Nouveau et entrez un nom descriptif

et les paramètres de la projection. Après cela vous pouvez sauver votre projection en cliquant le bouton Sauver.

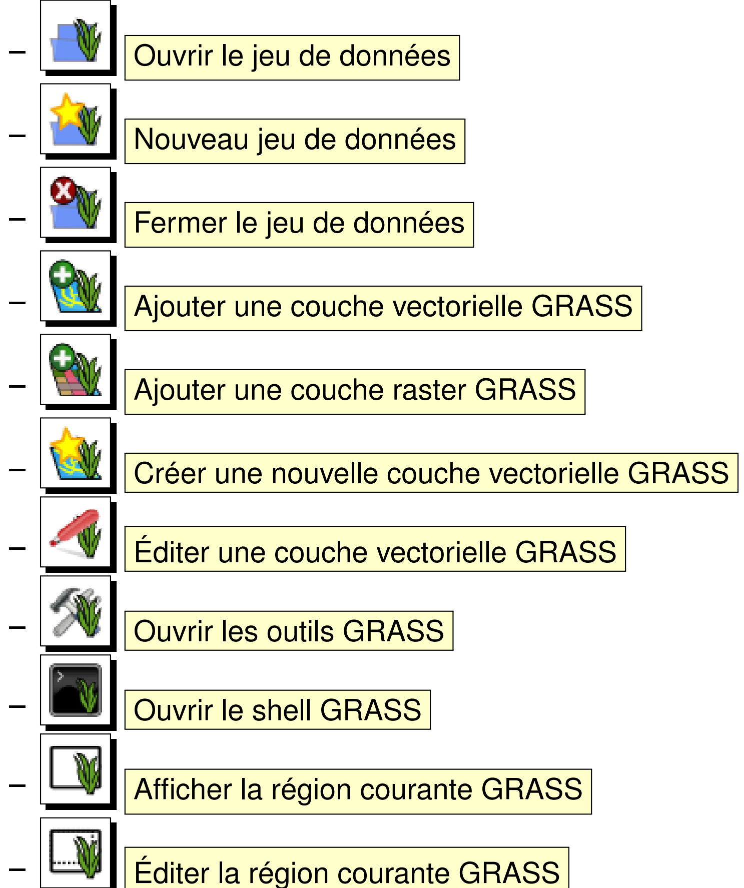
Remarquez que les Paramètres doivent débuter par un bloc `+proj=` pour représenter le nouveau système de coordonnées de référence.

Vous pouvez tester vos paramètres de projection pour voir s'il donne les mêmes résultats en cliquant sur le bouton Calculer dans le bloc Test et copiant vos paramètres de projection dans le champ Paramètres. Puis entrez une latitude et longitude connue en WGS 84 dans les champs Nord et Est respectivement. Cliquez sur le bouton Calculer et comparez les résultats avec les valeurs connues dans votre système de coordonnées de référence.

9 Intégration du SIG GRASS

L'extension GRASS fourni un accès aux bases de données et aux fonctionnalités de GRASS. Cela inclut la visualisation des couches d'informations GRASS raster et vecteur, la numérisation de couches vecteurs, l'édition des attributs des couches d'informations vecteurs, la création de nouvelles couches et l'analyse 2D et 3D grâce à l'accès à près de 300 modules GRASS

Dans cette section, nous présenterons les fonctionnalités de l'extension et nous donnerons des exemples sur la manière de gérer et de travailler avec des données GRASS. Les fonctionnalités principales suivantes sont fournies dans la barre de menu lorsque vous lancez l'extension GRASS, comme décrit dans la section 9.1 :



- Ouvrir le jeu de données
- Nouveau jeu de données
- Fermer le jeu de données
- Ajouter une couche vectorielle GRASS
- Ajouter une couche raster GRASS
- Créer une nouvelle couche vectorielle GRASS
- Éditer une couche vectorielle GRASS
- Ouvrir les outils GRASS
- Ouvrir le shell GRASS
- Afficher la région courante GRASS
- Éditer la région courante GRASS

9.1 Lancer l'extension GRASS

Pour pouvoir utiliser les fonctionnalités de GRASS et/ou visualiser des données vecteurs ou raster dans QGIS, vous devez sélectionner et charger l'extension GRASS à l'aide du gestionnaire d'extensions. Pour le faire cliquez sur le menu Plugins > Gestionnaire de Plugins, sélectionnez GRASS et cliquez sur OK.

Vous pouvez maintenant charger des données raster et vecteur depuis un SECTEUR GRASS existant (voir Section 9.2). Ou alors vous pouvez créer un nouveau SECTEUR GRASS à l'aide de QGIS (voir Section 9.3.1) et y importer des données raster et vecteur (voir Section 9.4) pour réaliser des traitements à l'aide de la boîte à outils GRASS.9.9).

9.2 Charger des données GRASS raster et vecteur

Avec l'extension GRASS, vous pouvez charger des données raster ou vecteur l'aide du bouton approprié dans la barre de menu. Ici nous utiliserons comme exemple, le jeu de données QGIS Alaska (voir Section 3.2). Il contient un SECTEUR GRASS avec 3 couches vecteurs et 1 raster d'élévation.

1. Créez un nouveau répertoire `grassdata`, téléchargez le jeu de données QGIS alaska `qgis_sample_data.zip` depuis `http://download.osgeo.org/qgis/data/` et décompressez le dans le répertoire `grassdata`
2. Démarrez QGIS
3. Si cela n'a pas déjà été fait dans une précédente session QGIS, chargez l'extension GRASS en cliquant sur Plugins > Gestionnaire de Plugins et sélectionnez GRASS. La barre d'outils GRASS apparaît dans la barre de menu.
4. Dans la barre d'outils GRASS, cliquez sur le bouton Ouvrir le jeu de données pour ouvrir le gestionnaire de `Base de données`.
5. Pour `Base de données GIS` parcourez puis sélectionnez ou entrez le chemin vers le répertoire nouvellement créé, `grassdata`.
6. Vous devriez maintenant être capable de sélectionner le SECTEUR `alaska` et le jeu de données `demo`.
7. Cliquez sur (OK). Notez que les outils GRASS sont maintenant accessibles dans la barre d'outils.
8. Cliquez sur Ajouter une couche raster GRASS, choisissez le fichier `gtopo30` et cliquez sur OK. Vous visualisez alors la couche d'élévation.
9. Cliquez sur Ajouter une couche vectorielle GRASS, choisissez la couche `alaska` et cliquez sur OK. La couche vectorielle alaska s'affiche au-dessus du raster gtopo30. Vous pouvez modifier les propriétés de la couche d'information comme décrit dans le chapitre 5.3. Vous pouvez par exemple modifier la transparence, changer la couleur du contour ou celle du remplissage.
10. Chargez également les deux autres couches vecteur `rivers` et `airports` et modifiez leurs propriétés.

Comme vous pouvez le constater, il est très facile d'afficher des données GRASS raster et vecteur dans QGIS. Dans les sections suivantes, nous allons voir comment éditer des données GRASS et créer un nouveau SECTEUR. Vous trouverez sur le site GRASS http://grass.osgeo.org/download/data.php d'autres exemples de SECTEURS.

Astuce 32 CHARGEMENT DE DONNÉES GRASS
Si vous rencontrez des problèmes lors du chargement de données ou si QGIS se ferme anormalement, vérifiez que vous que avez bien charger l'extension GRASS comme décrit dans la section 9.1.

9.3 Secteur et Jeu de données GRASS

Les données GRASS sont stockées dans un répertoire référencé sous le nom GISDBASE. Ce répertoire, souvent appelé grassdata, doit être créé avant que vous commenciez à travailler avec l'extension GRASS dans QGIS. Dans ce répertoire, les données GRASS sont organisées par projets et stockées dans des sous-répertoires appelés SECTEUR (LOCATION). Chaque SECTEUR est définit par son système de coordonnées, sa projection et son étendue géographique. Chaque SECTEUR peut contenir plusieurs Jeux de données (MAPSETs) (sous-répertoires de SECTEUR) qui sont utilisés pour subdiviser le projet en différents thèmes, sous-regions ou espaces de travail pour chaque membre d'une équipe.(Neteler & Mitasova 2008 **?**). Pour pouvoir analyser des couches raster ou vecteur à l'aide des modules GRASS, vous devez les importer dans un SECTEUR. [5]

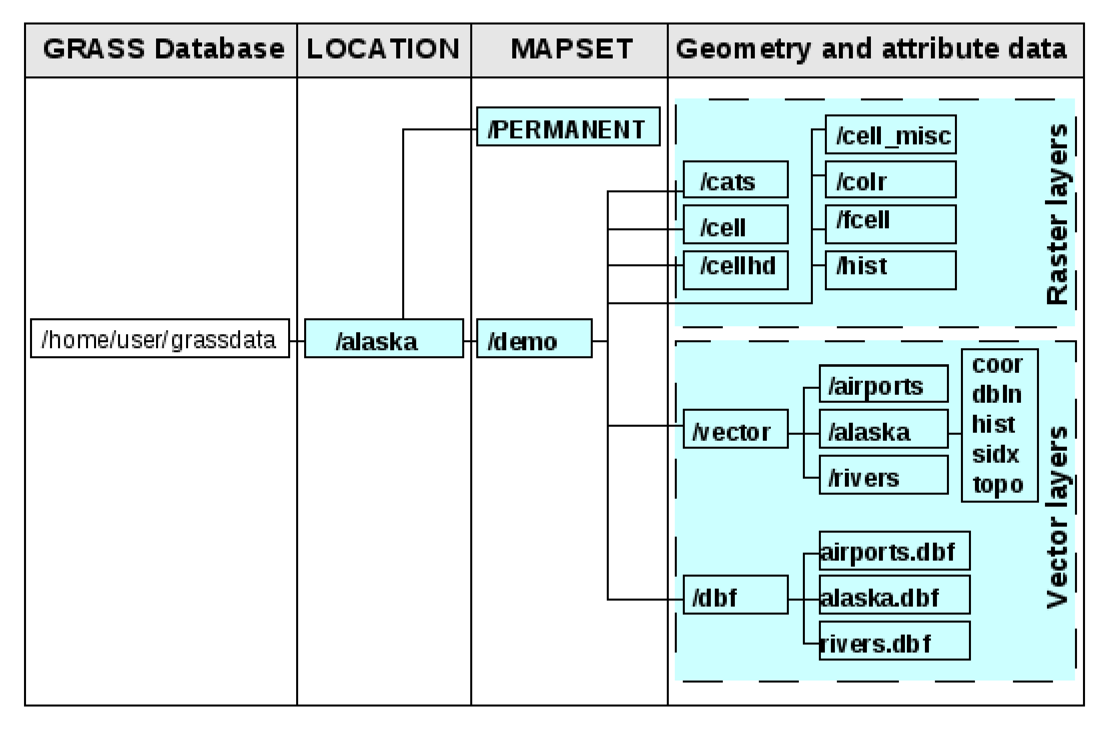
9.3.1 Créer un nouveau SECTEUR GRASS

A titre d'exemple, vous trouverez des instructions sur la manière dont le SECTEUR alaska échantillon, qui est projeté en Albers Equal Area et ayant pour unité le pied, a été créé pour l'échantillon de données QGIS. Ce SECTEUR alaska échantillon sera utilisé pour tous les exemples et exercices des chapitres GRASS qui suivent. Il est utile de le télécharger et de l'installer 3.2).

1. Démarrez QGIS et assurez vous que l'extension GRASS est chargée.
2. Affichez le shapefile alaska.shp (voir Section 5.1.1) du jeu de données QGIS alaska 3.2.
3. Dans la barre d'outils GRASS, cliquez sur Nouveau jeu de données pour ouvrir l'assistant Jeux de données.
4. Sélectionnez un répertoire existant de base de données GRASS (Base de données) grassdata ou créez en un pour le nouveau SECTEUR avec le gestionnaire de fichiers de votre ordinateur. Cliquez le bouton Suivant .

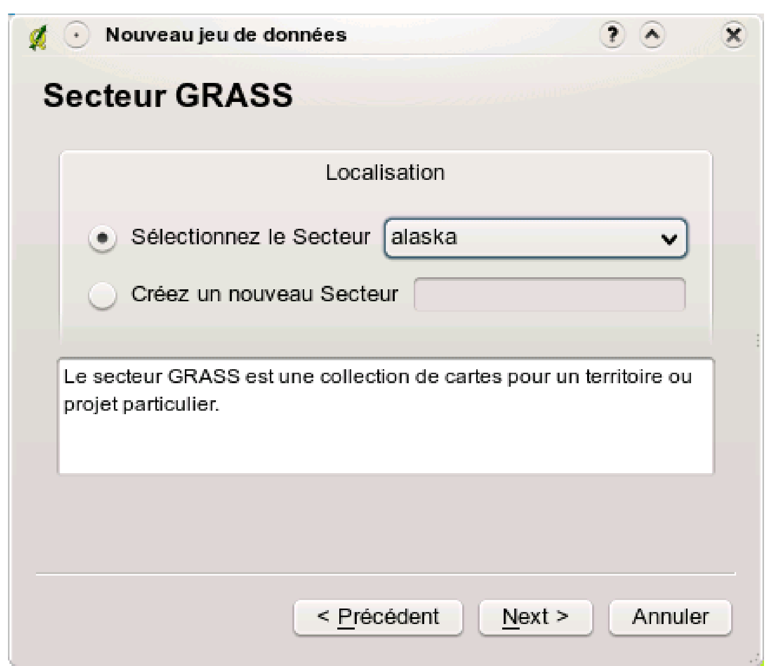
[5] Ce n'est pas complètement vrai car avec les modules GRASS r.external et v.external, vous pouvez lier (en lecture seule) des jeux de données externes sans les importer. Ces jeux de données doivent être supportés par la librairie GDAL/OGR. Mais comme il ne s'agit pas d'une fonctionnalité courante pour les débutants sur GRASS, elle ne sera pas décrite ici.

FIG. 19: Données GRASS dans le SECTEUR alaska (adapté de Neteler & Mitasova 2008 **?**)



5. Nous pouvons utiliser cet assistant à la fois pour créer un nouveau `Jeu de données` dans un `SECTEUR` existant (voir Section 9.3.2) et pour créer un nouveau `SECTEUR`. Cliquez sur le bouton radio ⊙ Créez un nouveau secteur (voir Figure 20).
6. Entrez un nom pour le `SECTEUR` - nous utilisons alaska et cliquez sur le bouton [Suivant]
7. Définissez la projection en cliquant sur le bouton radio ⊙ Projection pour activer la liste des projections
8. Nous utilisons la projection Albers Equal Area Alaska (pieds).Étant donné que nous savons qu'elle correspond au code EPSG 2964, nous le saisissons dans le champ de recherche. (Note : Si vous souhaitez reproduire la manipulation pour un autre `SECTEUR` et une autre projection et que vous ne connaissez pas le code EPSG, cliquez sur [Statut de la projection] dans le coin inférieur droit de la barre d'état (voir Section 8.3)).
9. Cliquez sur [Trouver] pour sélectionner la projection
10. Cliquer sur [Suivant]
11. Pour définir la région par défaut, nous devons saisir les limites Nord, Sud , Est et Ouest du `SECTEUR`. Ici il suffit de cliquer sur le bouton [Fixer l'emprise courante de QGIS] , pour appliquer l'emprise du shapefile `alaska.shp` déjà chargé comme emprise par défaut.
12. Cliquer sur [Suivant]

FIG. 20: Créer un nouveau SECTEUR ou JEU DE DONNEES GRASS dans QGIS



13. Nous avons aussi besoin de définir un `Jeu de données` dans notre nouveau `SECTEUR`. Vous pouvez l'appeler comme vous le souhaitez - nous utiliserons demo. [6]
14. Vérifiez le résumé pour vous assurez que tout est correct et cliquez sur [Terminer]
15. Le nouveau `SECTEUR alaska` et les deux `Jeux de données demo` et `PERMANENT` sont créés. Actuellement, le jeu de données courant est le `Jeux de données demo`, tel que vous l'avez défini.
16. Notez que certains outils qui n'étaient pas accessibles le sont maintenant.

Si cela semble faire beaucoup d'étapes, mais c'est en fait un moyen simple et rapide de créer un `SECTEUR`. Le `SECTEUR alaska` est maintenant prêt pour l'importation de données (voir Section 9.4). Vous pouvez également utiliser des données raster ou vecteur existantes dans le `SECTEUR alaska` inclues dans le jeu de données QGIS alaska 3.2 et continuez dans la section 9.5.

9.3.2 Ajouter un nouveau Jeu de données

Un utilisateur a seulement des droits d'écriture sur le `Jeu de données` GRASS qu'il a créé. Cela veut dire qu'au delà de l'accès à son propre `Jeu de données` GRASS, chaque utilisateur peut aussi lire

[6] Quand nous créons un nouveau `SECTEUR`, GRASS créé automatiquement un `Jeu de données` spécial appelé `PERMANENT` conçu pour stocker les données essentiels du projet, l'extension spatiale par défaut et la définition du système de coordonnées (Neteler & Mitasova 2008 **?**).

les données dans les autres `Jeux de données`, mais il ne peut modifier et supprimer que les données que dans son propre `Jeu de données`. Tous les `Jeux de données` incluent un fichier `WIND` qui stocke l'extension et la résolution raster courante (Neteler & Mitasova 2008 **?**, voir Section 9.8).

1. Démarrer QGIS et assurez vous que l'extension GRASS est chargée
2. Dans la barre d'outils GRASS, cliquez sur Nouveau jeu de données pour ouvrir l'assistant `Jeux de données`.
3. Sélectionnez le répertoire `grassdata` de la base de données GRASS (Base de données) qui contient déjà le `SECTEUR alaska` et où nous voulons ajouter un autre `SECTEUR` appelé test.
4. Cliquez sur Suivant .
5. Nous pouvons utiliser cet assistant à la fois pour créer un nouveau `Jeu de données` dans le `SECTEUR` existant et pour créer un nouveau `SECTEUR`. Cliquez sur le bouton radio ⊙ Sélectionnez le Secteur (voir Figure 20) et cliquez sur Suivant .
6. Entrez le `text` du nom pour le nouveau `Jeu de données`. En dessous, dans l'assistant, vous pouvez voir une liste des `Jeux de données` et de leurs propriétaires.
7. Cliquez sur Suivant , vérifiez le résumé pour vous assurer qu'il est correct et cliquez sur Terminer

9.4 Importer des données dans un SECTEUR GRASS

Cette section donne un exemple d'importation de données raster et vecteur dans le `SECTEUR` GRASS `alaska` fournit dans le jeu de données QGIS alaska. Nous utiliserons une couche raster d'occupation du sol `landcover.img` et une couche vecteur au format GML `lakes.gml`, toutes deux présentes dans le jeu de données alaska.

1. Démarrer QGIS et assurez vous que l'extension GRASS est chargée.
2. Dans la barre d'outil GRASS, cliquez sur Ouvrir un jeu de données pour ouvrir l'assistant `Jeu de données`.
3. Sélectionnez comme base de données GRASS le répertoire `grassdata` dans le jeu de données QGIS alaska, comme `SECTEUR alaska`, comme as `Jeu de donnée demo` et cliquez sur OK .
4. Maintenant cliquez sur Ouvrir les outils GRASS. La boîte à outils GRASS (voir Section 9.9) s'ouvre.
5. Pour importer la couche raster `landcover.img`, cliquez sur le module `r.in.gdal` dans l'onglet Arborescence des modules . Ce module GRASS vous permet d'importer les fichiers raster supportés par la librairie GDAL dans un `SECTEUR` GRASS. La boîte de dialogue `r.in.gdal` apparaît.

6. Naviguer jusqu'au répertoire `raster` dans le jeu de données QGIS alaska et sélectionnez le fichier `landcover.img`.
7. Définissez `landcover_grass` comme nom de sortie et cliquez sur [Lancer]. Dans l'onglet [Rendu], vous voyez la commande GRASS en cours `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass`.
8. Lorsque **Terminé avec succès** s'affiche cliquez sur [Vue]. La couche raster `landcover_grass` est maintenant importée dans GRASS et pourra être affichée dans QGIS.
9. Pour importer le fichier GML `lakes.gml`, cliquez sur le module `v.in.ogr` dans l'onglet [Arborescence des modules].La boîte de dialogue `v.in.ogr` apparaît.
10. Naviguer jusqu'au répertoire `gml` dans le jeu de données QGIS alaska et sélectionnez le fichier `lakes.gml`.
11. Définissez `lakes_grass` comme nom de sortie et cliquez sur [Lancer]. Vous n'avez pas besoin des autres options dans cet exemple.
12. Lorsque **Terminé avec succès** s'affiche cliquez sur [Vue]. Le fichier `lakes_grass` est maintenant importé dans GRASS et pourra être affiché dans QGIS.

9.5 Le modèle vecteur de GRASS

Il est important de comprendre le modèle vecteur de GRASS avant de faire de la numérisation. En général, GRASS utilise un modèle vecteur topologique. Cela veut dire que les surfaces ne sont pas représentées par des polygones fermés mais par une ou plusieurs limites. Une limite entre des polygones adjacents n'est numérisée qu'une seule fois et est partagée par les deux surfaces. Les limites doivent être connectées sans trous. Une surface est identifiée (libellée) via le centroide de la surface.

Outre les limites et centro"ides, une couche vecteur peut également contenir des points et des lignes. Tous ces éléments de géométrie peuvent être mélangés dans une couche vecteur et seront représentés dans différentes «sous-couches» dans une carte vectorielle GRASS. Ainsi, une couche GRASS n'est pas un vecteur ou un raster, mais un niveau à l'intérieur d'une couche vecteur. Il est important de bien distinguer ceci.

7

Il est possible de stocker plusieurs sous-couches dans une couche vecteur. Par exemple, des champs, de la forêt et des lacs peuvent être stockés dans une couche vecteur. Les forêts et les lacs adjacents partagent les mêmes limites, mais ils auront des tables attributaires différentes. Il est aussi possible de faire correspondre une table attributaires aux limites. Par exemple, la limite entre un lac et une forêt peut être une route qui peut avoir une table attributaire différente.

[7] Même s'il est possible de mélanger des éléments de géométries différentes, c'est inhabituel et même dans GRASS, on l'utilise dans des cas particuliers tel que l'analyse de réseau. Normalement, vous devriez stocker des éléments de géométries différentes dans des couches différentes

La 'sous-couche' est définie dans GRASS par un chiffre. Ce chiffre définit s'il y a plusieurs sous-couches à l'intérieur d'une couche vecteur. Par exemple, il définit s'il s'agit de lac ou de forêt. Pour l'instant, il s'agit d'un nombre mais dans des versions futures GRASS pourra utiliser des noms pour les sous-couches dans l'interface utilisateur.

Les données attributaires peuvent être stockées dans le `SECTEUR` au format DBase ou SQLIte3 ou dans des tables de bases de données externes comme par exemple : PostgreSQL, MySQL, Oracle, etc.

Les données attributaires sont liées à la géométrie par le biais d'un champ 'category'. 'Category' (clé, ID) est un entier attaché à la géométrie, et il est utilisé comme lien vers une colonne de clé dans la table de base de données.

Astuce 33 APPRENDRE LE MODÈLE VECTEUR DE GRASS
Le meilleur moyen d'apprendre le modèle vecteur de GRASS et ses possibilités est de télécharger un des nombreux tutoriels GRASS où le modèle vecteur est décrit plus précisément. Voir `http://grass.osgeo.org/gdp/manuals.php` pour des informations complémentaires, des livres et des tutoriels dans différentes langues.

9.6 Création d'une nouvelle couche vecteur GRASS

Pour créer une nouvelle couche vecteur GRASS à l'aide de l'extension GRASS, cliquez sur  Créez une nouvelle couche vectorielle GRASS dans la barre d'outils. Entrez le nom dans la boîte de dialogue et vous pouvez commencer à digitaliser un point une ligne ou un polygone en suivant les instructions de la section 9.7. Dans GRASS, il est possible de gérer plusieurs types de géométrie (point, ligne et surface) dans une seule couche d'information, car GRASS utilise un modèle vecteur topologique. Vous n'avez donc pas besoin de sélectionner un type de géométrie quand vous créez une couche vecteur GRASS. C'est différent de la création de shapefile avec QGIS, car les shapefiles utilisent un modèle vecteur d'entité simple (Simple Feature vector model) (voir Section 5.4.4).

Astuce 34 CRÉATION D'UNE TABLE ATTRIBUTAIRE POUR UNE NOUVELLE COUCHE VECTEUR GRASS
Si vous souhaitez renseigner les données attributaires de vos entités numérisées, assurez-vous d'avoir créé une table attributaire avec des champs avant ce commencer votre numérisation (voir Figure 25).

9.7 Numérisation et édition de couche vecteur GRASS

Les outils de numérisation pour les couches vecteurs de GRASS sont accessibles via  Éditer une couche vectorielle GRASS dans la barre d'outils. Assurez-vous d'avoir ouvert une couche vectorielle GRASS ainsi que d'avoir sélectionné la sous-couche dans la légende avant d'utiliser l'outil d'édition. La figure 22 montre la boîte de dialogue GRASS qui s'affiche quand vous cliquez

sur l'outil d'édition. Les outils et les paramètres d'édition sont détaillés dans les sections suivantes.

Astuce 35 NUMÉRISATION DE POLYGONES DANS GRASS
Si vous voulez créer un polygone dans GRASS, vous devez numériser premièrement les limites du polygone, en définissant mode sur `Pas de catégorie`.Ensuite, vous ajoutez un centro"ide (emplacement de l'étiquette) dans le polygone fermé, fixant le mode sur `Prochain non utilisé`. La raison en est, que le modèle vectoriel topologique assure toujours le lien entre les informations d'attributs des polygones via le centro"ide et non via la limite.

Barre d'outils

Sur la figure 21 vous pouvez voir la barre d'outils d'édition GRASS de l'extension GRASS. Le tableau 4 récapitule les fonctions disponibles.

FIG. 21: Barre d'outils d'édition GRASS



Onglet Categorie

L'onglet Categorie vous permet de définir la manière dont les valeurs du champ category sont assignées au nouvel élément géométrique.

FIG. 22: Onglet Catégorie d'Édition GRASS



- **Mode** : quelle catégorie sera appliqué au nouvel élément.
  - Prochain non utilisé - applique la valeur suivante non utilisée du champ category à l'élément géométrique.
  - Saisie manuelle - saisir manuellement la valeur du champ category pour l'élément géométrique.
  - Pas de catégorie - ne pas remplir le champ category. C'est par exemple utilisé pour les surfaces car les valeurs de catégorie sont stockées via le centro"ide

TAB. 4: Outils de numérisation GRASS

Icône	Outil	Fonction
	Nouveau Point	Numérise un nouveau point
	Nouvelle Ligne	Numérise une nouvelle ligne (terminez la numérisation en sélectionnant un nouvel outil)
	Nouveau Contour	Numérise un nouveau contour (terminer la numérisation en sélectionnant un nouvel outil)
	Nouveau Centro"ide	Numérise un nouveau centro"ide (emplacement de l'étiquette d'un polygone existant)
	Déplacer le sommet	Déplace un sommet d'une ligne ou d'un polygone existant et indique sa nouvelle position
	Ajouter un sommet	Ajoute un nouveau sommet à une ligne existante
	Effacer un sommet	Efface un sommet d'une ligne existante (confirmez le sommet sélectionné avec un autre clic)
	Déplace l'élément	Déplacez la limite, la ligne, le point ou le centro"ide sélectionné puis cliquez sur la nouvelle position
	Coupe la ligne	Coupe une ligne existante en deux parties
	Efface l'élément	Efface une limite, une ligne, un point ou un centro"ide existant (confirmez l'élément sélectionné avec un autre clic)
	Éditer les attributs	Édite les attributs de l'élément sélectionné (notez qu'un seul élément peut représenter plusieurs géométries, voir ci-dessus)
	Fermer	Ferme la session et sauvegarde l'état actuel (reconstruit la topologie après)

– **Categorie** - un identifiant (ID) est attaché à chaque objet numérisé. Il est utilisé pour connecter les objets géométriques avec ces attributs.
– **Couche** - Chaque objet peut être connecté à différentes tables attributaires au travers des différentes sous-couches. Le numéro de sous-couche par défaut est 1.

Astuce 36 CRÉATION D'UNE SOUS-COUCHE SUPPLÉMENTAIRE AVEC QGIS

Si vous souhaitez avoir plusieurs sous-couches dans votre couche vecteur, ajouter simplement un nouveau chiffre dans la zone de saisie 'Couche' et appuyez sur entrée. Dans l'onglet Table, vous pouvez créer de nouvelles tables attributaires connectées à votre nouvelle sous-couche.

Onglet Paramètres

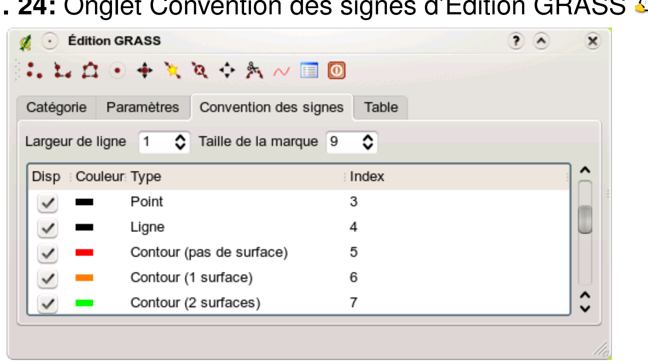
L'onglet Paramètres vous permet de définir la tolérance d'accrochage en pixels écran. Le seuil définit à partir de quelle distance les nouveaux points ou les nouvelles lignes sont accrochées automatiquement à des noeuds existants. Cela aide à éviter de créer des trous ou des superpositions entre les contours. La valeur par défaut est fixée à 10 pixels.

FIG. 23: Onglet Paramètres d'Édition GRASS

Onglet Convention des signes

L'onglet Convention des signes vous permet d'afficher et modifier la symbologie, la couleur des différentes formes géométriques ainsi que leur statut topologique (par exemple : contour ouvert / fermé)

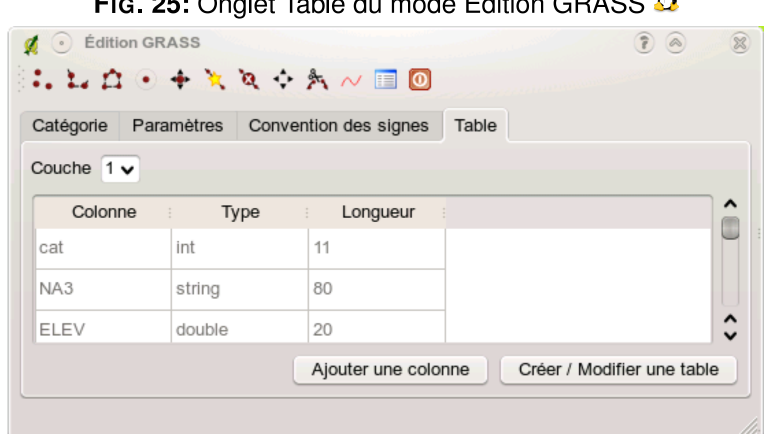
FIG. 24: Onglet Convention des signes d'Édition GRASS



Onglet Table

L'onglet Table donne des informations sur la table attributaire d'une sous-couche donnée. C'est ici que vous pouvez ajouter des colonnes à une table attributaire existante ou créer une nouvelle table attributaire pour une nouvelle couche vectorielle GRASS (voir Section 9.6).

FIG. 25: Onglet Table du mode Édition GRASS



Astuce 37 ÉDITER LES PERMISSIONS GRASS

Vous devez être propriétaire du `Jeu de données` que vous voulez éditer. Il est impossible de modifier des informations d'un `Jeu de données` qui n'est pas à vous, même si vous avez des droits en écriture.

9.8 L'outil région GRASS

La définition d'une région (définir une emprise spatiale de travail) dans GRASS est très importante pour travailler avec des couches rasters. Le travail d'analyse vecteur n'est, par défaut, pas limitée à une région définie. Tous les rasters nouvellement créés auront l'extension spatiale et la résolution de la région GRASS en cours d'utilisation, indépendamment de leur extension et résolution d'origine. La région courante GRASS est stockée dans le fichier `$LOCATION/$MAPSET/WIND`, et celui-ci définit les limites Nord, Sud, Est et Ouest, le nombre de lignes et de colonnes ainsi que la résolution spatiale horizontale et verticale.

Il est possible de d'afficher ou de masquer l'affichage de la région GRASS dans QGIS à l'aide du bouton Afficher la région courante GRASS . .

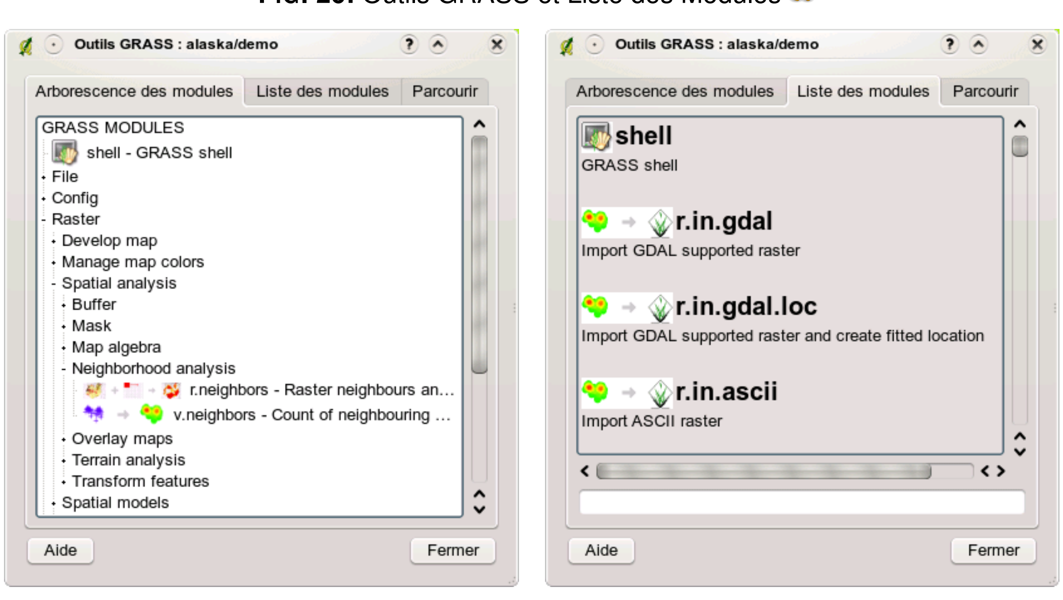
A l'aide du bouton Éditer la région courante GRASS vous avez accès à une boîte dialogue qui vous permet de modifier la région courante ainsi que sa symbologie. Entrez les nouvelles limites et résolution et cliquez sur OK . Cette boîte de dialogue vous permet aussi de définir une nouvelle région interactivement à l'aide de la souris. Pour définir ce rectangle d'emprise, cliquez avec le bouton gauche de la souris et définissez un rectangle que vous terminerez en cliquant de nouveau sur le bouton gauche de la souris et fermez la boîte de dialogue en cliquant sur OK . Le module GRASS `g.region` propose un grand nombre de paramètres pour définir de façon appropriée les limites et la résolution d'une région pour faire de l'analyse raster. Vous pouvez vous servir de ces paramètres dans la boîte à outils GRASS décrite dans la section 9.9.

9.9 La boîte à outils GRASS

La boîte de dialogue [Ouvrir les outils GRASS] donne accès aux fonctionnalités GRASS qui permettent de travailler dans un SECTEUR et sur un Jeu de Données. Pour utiliser les outils GRASS, vous devez ouvrir un SECTEUR et un Jeu de Données sur lequel vous avez des droits d'écriture (que vous avez normalement si vous avez créé le Jeu de Données). Cela est nécessaire car les rasters et les vecteurs nouvellement créés lors des analyses doivent être écrits dans le SECTEUR et Jeu de Données courant.

9.9.1 Travailler avec les modules GRASS

FIG. 26: Outils GRASS et Liste des Modules



(a) Arborescence des modules

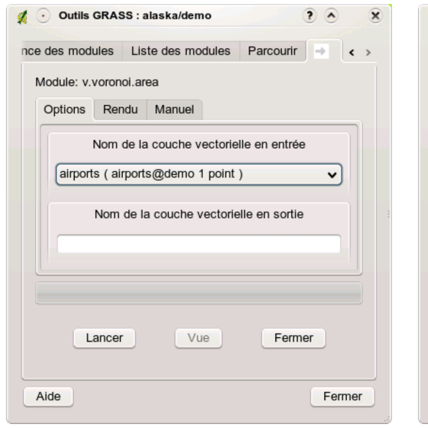
(b) Liste des modules avec possibilité de recherche

L'invite de commande de la boîte à outils GRASS vous donne accès à pratiquement tous les modules GRASS (près de 300) en ligne de commande. Afin d'offrir un environnement de travail plus agréable, environ 200 d'entre eux sont fournis avec une boîte de dialogue. Ces boîtes de dialogue sont groupées par thèmes mais sont aussi accessibles par une recherche libre. Vous trouverez une liste complète des modules GRASS disponibles dans QGIS 1.0.0 dans l'annexe B. Il est aussi possible de personnaliser le contenu de la boîte à outils GRASS. Ceci est décrit dans la section 9.9.3.

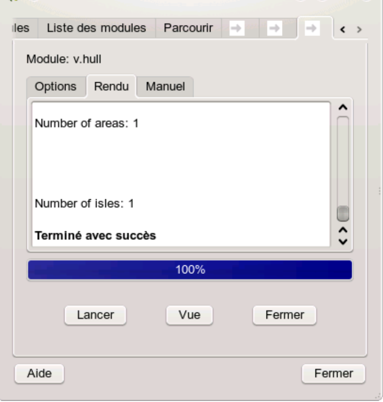
Comme indiqué sur la figure 26, vous pouvez chercher le module GRASS approprié en utilisant l'onglet [Arborescence des modules] ou en utilisant l'onglet [Liste des Modules] pour faire une recherche.

Lorsque vous cliquez sur un module, un nouvel onglet apparaît proposant trois sous-onglets Options , Rendu et Manuel . Sur la figure 27, vous voyez un exemple pour le module GRASS `v.buffer`.

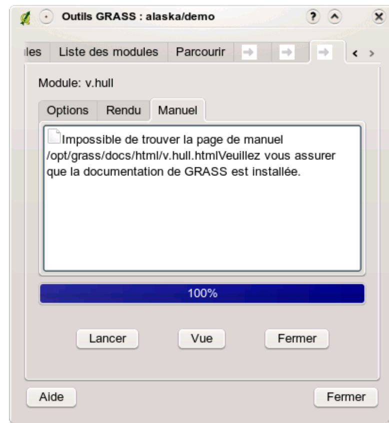
FIG. 27: Boîte de dialogue d'un module issue des outils GRASS



(a) Options du module



(b) Rendu du module



(c) Aide sur le module

Options

L'onglet Options propose une interface simplifiée où vous pouvez sélectionner un raster ou un vecteur en cours de visualisation dans QGIS et saisir les paramètres spécifiques au module avant de le lancer. Tous les paramètres du module ne sont généralement pas fournis afin de simplifier les boîtes de dialogue. Pour utiliser des paramètres que ne se trouvent pas dans la boîte de dialogue, vous devez utiliser l'invite de commande et lancer les modules en lignes de commande.

Rendu

L'onglet Rendu fournit des informations sur l'état de sortie du module. Quand vous cliquez sur le bouton Lancer , le module passe sur l'onglet Rendu et vous voyez les informations sur le processus en cours. Si tout se passe bien, vous verrez finalement le message `Terminé avec succès`.

Manuel

L'onglet Manuel montre la page HTML d'aide du module. Vous pouvez vous en servir pour voir les autres paramètres du modules et pour avoir une connaissance plus approfondie de l'objet du module. A la fin de chaque page d'aide d'un module, vous avez des liens vers `Main Help index`, `Thematic index` et `Full index`. Ces liens vous donnent les mêmes informations que si vous utilisiez `g.manual`.

Astuce 38 AFFICHER LES RÉSULTATS IMMÉDIATEMENT
Si vous voulez voir immédiatement dans votre fenêtre carte le résultat du module, vous pouvez utiliser le bouton 'Vue' au bas de l'onglet du module.

9.9.2 Travailler avec le navigateur GRASS

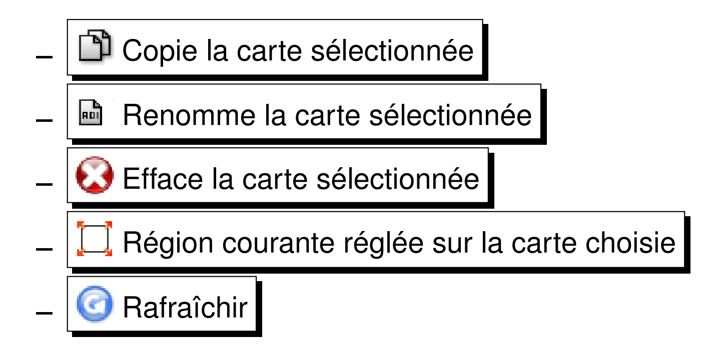
Une autre fonctionnalité utile dans la boîte à outils GRASS est le navigateur de SECTEUR GRASS. Sur la figure 28 vous pouvez voir le SECTEUR en cours avec ses Jeux de Données.

Dans la partie gauche de la fenêtre vous pouvez naviguer dans tous les Jeux de Données du SECTEUR courant. La partie droite de la fenêtre affiche des informations sur le raster ou le vecteur sélectionné, tel que la résolution, l'emprise, la source des données, les tables attributaires pour les vecteurs et un historique des commandes.

FIG. 28: Navigateur de SECTEUR GRASS

La barre d'outils de l'onglet Parcourir donne accès à des outils de gestion du SECTEUR sélectionné :

– Ajoute la carte sélectionnée à la carte QGIS



Les commandes Renomme la carte sélectionnée et Efface la carte sélectionnée ne fonctionnent qu'avec les cartes présente dans votre `Jeu de Données` sélectionné. Tous les autres outils fonctionnent aussi avec les autres `Jeux de Données`.

9.9.3 Personnaliser la boîte à outils GRASS

Pratiquement tous les modules GRASS peuvent être ajoutés à la boîte à outils. Une interface XML est fournie pour analyser les fichiers XML très simple qui configurent l'apparence et les paramètres des modules dans la boîte à outils.

Un exemple de fichier XML pour le module `v.buffer` (v.buffer.qgm) est donné ci-dessous :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE qgisgrassmodule SYSTEM "http://mrcc.com/qgisgrassmodule.dtd">

<qgisgrassmodule label="Vector buffer" module="v.buffer">
        <option key="input" typeoption="type" layeroption="layer" />
        <option key="buffer"/>
        <option key="output" />
</qgisgrassmodule>
```

L'analyseur lit cette définition et crée un nouvel onglet à l'intérieur de la boîte à outils lorsque vous sélectionnez le module. Une description plus détaillée pour ajouter des modules, changer le groupe des modules, etc. est disponible sur le wiki QGIS à l'adresse
`http://wiki.qgis.org/qgiswiki/Adding_New_Tools_to_the_GRASS_Toolbox`.

10 Composeur de carte

Le composeur de carte fournit des fonctionnalités croissantes de mise en page et d'impression. Il vous permet d'ajouter des éléments tels qu'un cadre de carte QGIS, une légende, une échelle graphique, des images et des étiquettes. Vous pouvez modifier la taille, grouper et positionner chaque élément et ajuster leurs propriétés pour créer votre mise en page. Le résultat peut être imprimé (ainsi qu'imprimé en Postscript et PDF), exporter dans un format d'image ou en SVG.[8] Voyez une liste d'outils dans le tableau 5 :

TAB. 5: Outils du Composeur de carte

Icône	Objectif	Icône	Objectif
	Exporter dans un format d'image		Exporter la composition en SVG
	Imprimer pu exporte r comme PDF ou Postscript		Zoom à l'étendue maximale
	Zoom in		Zoom out
	Rafraichie la vue		Ajouter une nouvelle carte à partir du cadre de carte de QGIS
	Ajouter une image au composeur de carte		Ajoute des étiquettes à la composition de carte
	Ajoute une nouvelle légende à la composition de carte		Ajoute une barre d'échelle graphique à la composition de carte
	Sélectionne/déplace les objets dans la composition de carte		Déplace le contenu dans un objet
	Groupe les objets de la composition de carte		Désolidaise les objets de la composition de carte
	Passe les objets par dessus dans la composition de carte		Passe les objets par dessous dans la composition de carte
	Déplace les objets sélectionnés tout en haut		Déplace les objets sélectionnés tout en bas

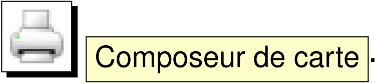
Pour accéder au composeur de carte, cliquez sur le bouton  dans la barre d'outils

[8]L'export en SVG est géré, mais il ne fonctionne pas proprement avec des versions récentes de QT4. Vous devez essayer et vérifier indivuellement sur votre système

ou choisissez Fichier > Composeur de carte.

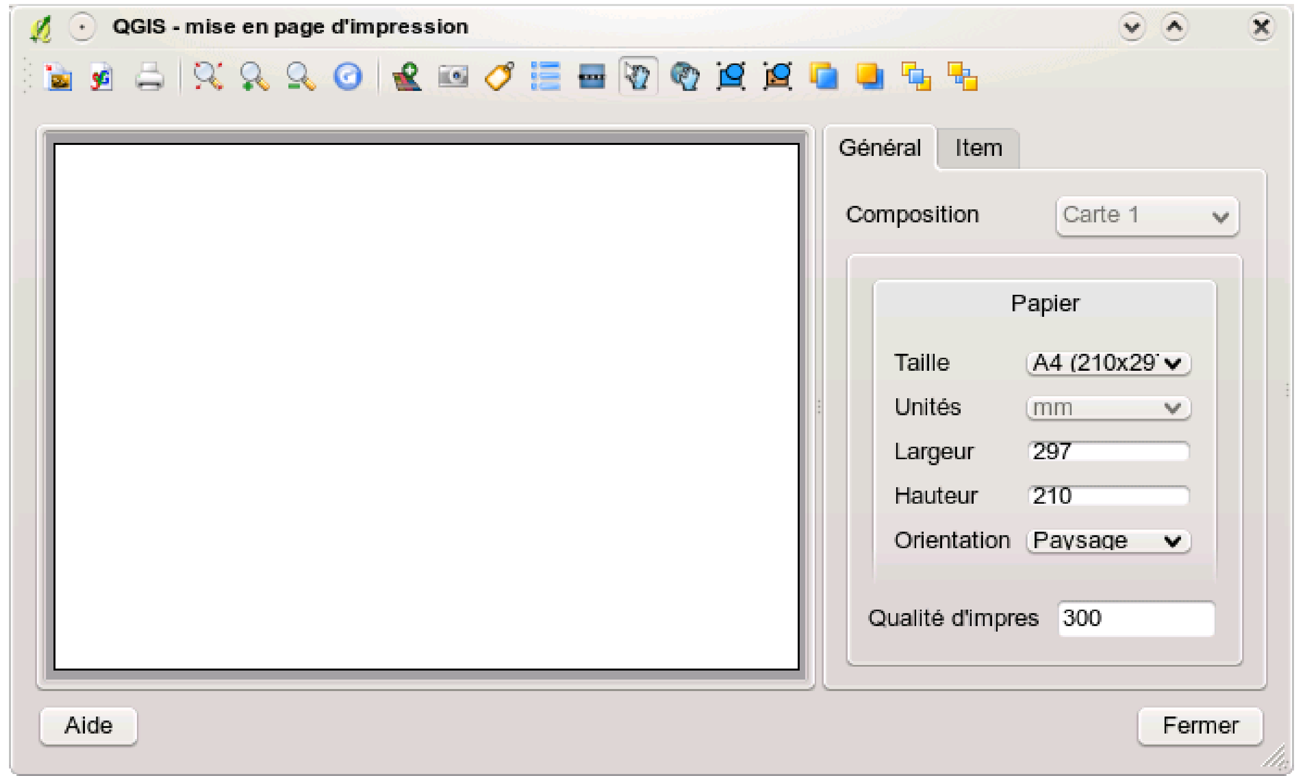
10.1 Utiliser le Composeur d'Impression

Avant de démarrer à travailler avec le composeur de carte, vous devez charger certaines couches raster et vecteurs dans la fenêtre de carte de QGIS et adapter leurs propriétés pour qu'elles vous conviennent. Après que tout soit rendu et symbolisé comme vous le souhaitez, cliquez sur l'icône



.

FIG. 29: Composeur de carte



Ouvrir le composeur de carte vous affiche un cadre vide auquel vous pouvez ajouter un cadre de la carte actuelle de QGIS, une légende, une échelle graphique, des images et du texte. La figure 29 montre la vue initiale du composeur de carte avant qu'un élément ne soit ajouté. Le composeur de carte affiche deux onglets :

– l'onglet Général vous permet de définir la taille du papier, l'orientation et la qualité d'impression pour le fichier de sortie, en dpi.
– L'onglet Item affiche les propriétés pour l'élément de la carte sélectionnée. Cliquez sur l'icône

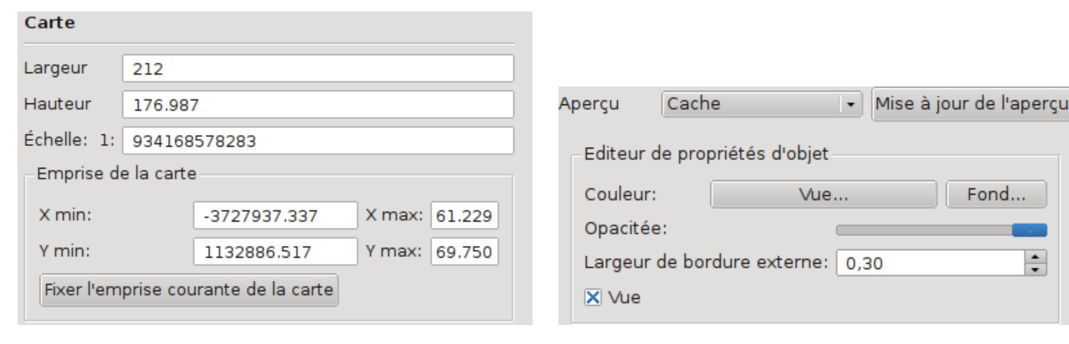
Sélectionner/Déplacer l'objet pour sélectionner un élément (par exemple l'échelle graphique ou une étiquette) dans le cadre. Puis cliquez sur l'onglet Item et personnalisez les paramètres pour l'élément sélectionné.

Vous pouvez ajouter de multiples éléments au composeur. Il est également possible d'avoir plus d'une vue de carte, légende ou échelle graphique dans le cadre du composeur de carte. Chaque élément possède ses propres propriétés et dans le cas de la carte, sa propre étendue.

10.1.1 Ajouter une carte en cours dans QGIS au Composeur d'Impression

pour ajouter le cadre de carte de QGIS, cliquez sur le bouton Ajouter une nouvelle carte à partir du cadre de carte de QGIS dans la barre d'outil du composeur de carte et dessiner un rectangle dans le cadre du composeur avec le bouton gauche de la souris pour ajouter la carte. Vous verrez une boîte vide avec un message *"La carte sera imprimée ici"*. Pour afficher la carte actuel, choisissez Preview Cache dans l'onglet Item de la carte.

FIG. 30: L'onglet Item de la carte dans le composeur de carte



(a) Width, height and extend dialog

(b) Boîte de dialogue des propriétés

Vous pouvez redimensionner la carte plus tard en cliquant sur le bouton Sélectioner/Déplacer l'objet, en sélectionnant l'élément, et en déplaçant un des curseurs bleus dans le coin de la carte. Avec la carte sélectionnée, vous pouvez maintenant adapter plus de propriétés dans l'onglet Item de la carte.

Pour déplacer une couche dans l'élément 'carte', sélectionnez l'élément 'carte', cliquez sur l'icône Déplacer le contenu de l'objet et déplacez les couches dans le cadre de l'élément 'carte' avec le bouton gauche de la souris.

Astuce 39 SAUVER UNE MISE EN PAGE DU COMPOSEUR DE CARTE

Si vous voulez sauver l'état actuel de la session du composeur de carte, cliquez sur Fichier > Sauvez le projet comme pour sauver l'état de votre espace de travail incluant l'état de la session actuelle du composeur de carte. Il est prévu, mais actuellement pas possible, de sauver les templates du composeur de carte eux-mêmes.

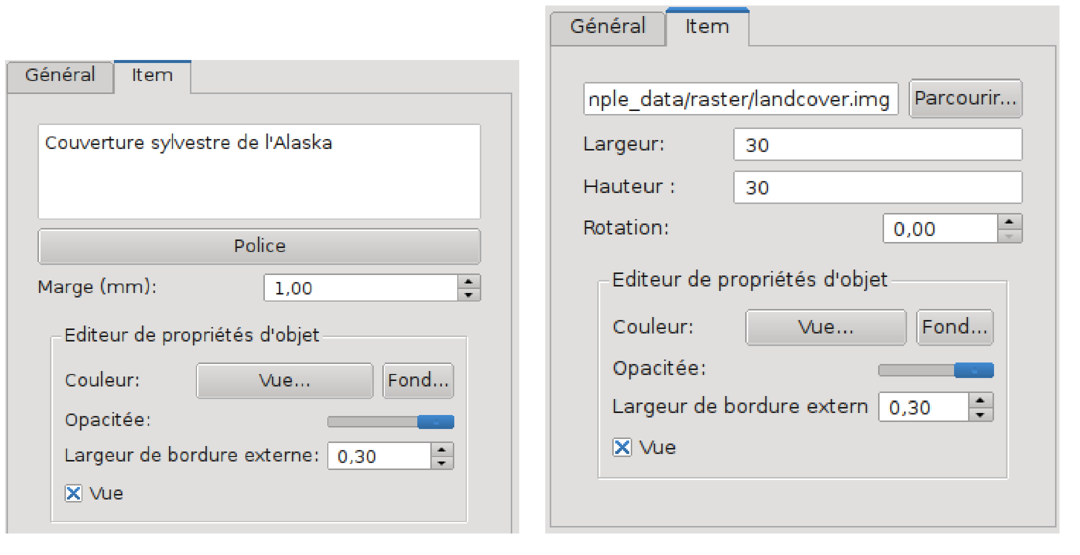
10.1.2 Ajouter d'autres éléments au Composeur d'Impression

Au-delà de l'ajout d'un cadre de la carte actuelle de QGIS au composeur de carte, il est également possible d'ajouter, déplacer et personnaliser les éléments légendes, échelles graphiques, images et étiquettes.

Étiquette et images

Pour ajouter une étiquette ou une image, cliquez sur l'icône Ajouter une étiquette ou Ajouter une image et placez l'élément avec le bouton gauche de la souris sur le cadre du composeur de carte.

FIG. 31: Personnaliser les étiquettes et les images du composeur de carte

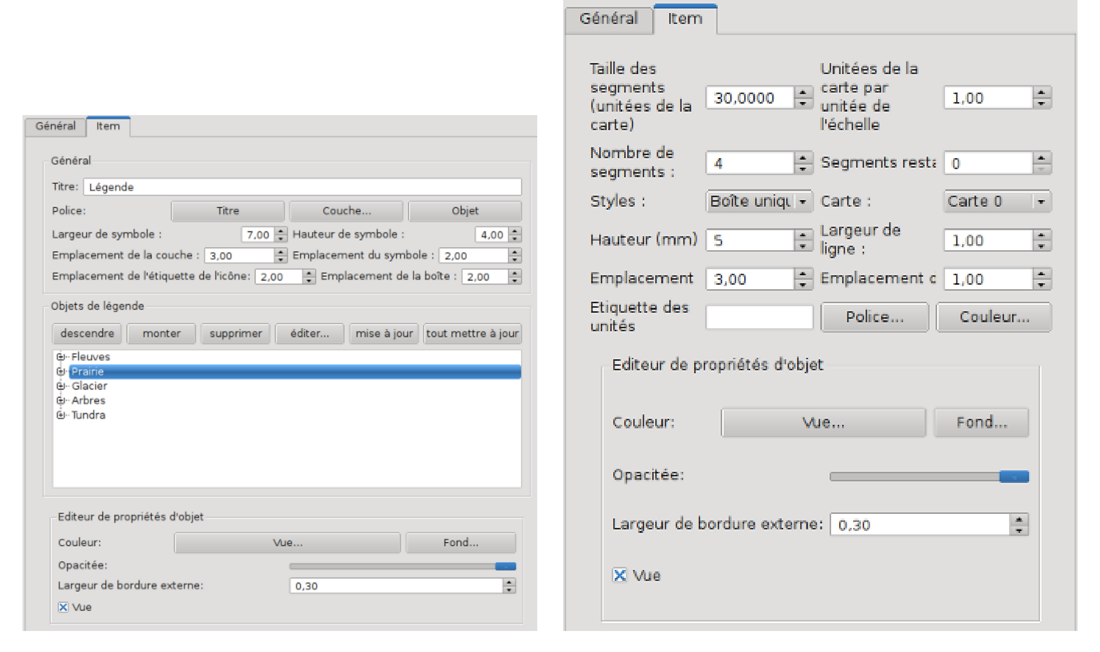


(a) onglet item des étiquettes (b) onglet item des images

Légende et barre d'échelle

Pour ajouter une légende ou une échelle graphique, cliquez sur l'icône  ajouter une nouvelle légende ou Ajouter une nouvelle échelle graphique et placez l'élément avec le bouton gauche de la souris sur le cadre du composeur de carte.

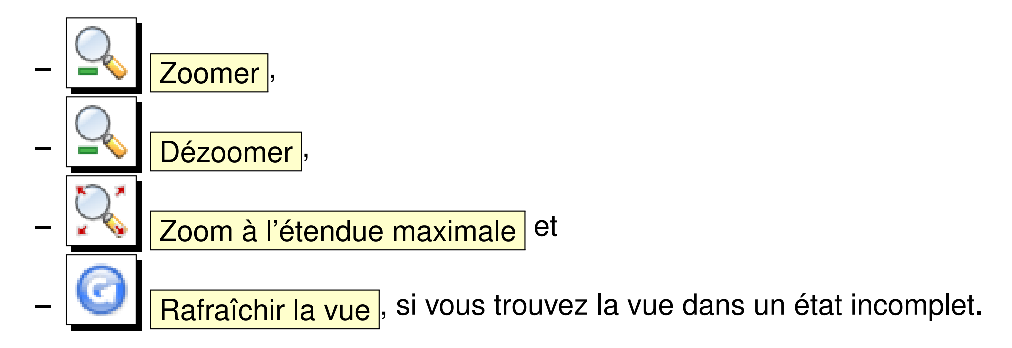
FIG. 32: Personnaliser la légende et l'échelle graphique du composeur de carte



(a) onglet item de la légende (b) onglet item de l'échelle graphique

10.1.3 Outils de navigation

Pour la navigation de la carte le composeur de la carte fournie 4 outils généraux :

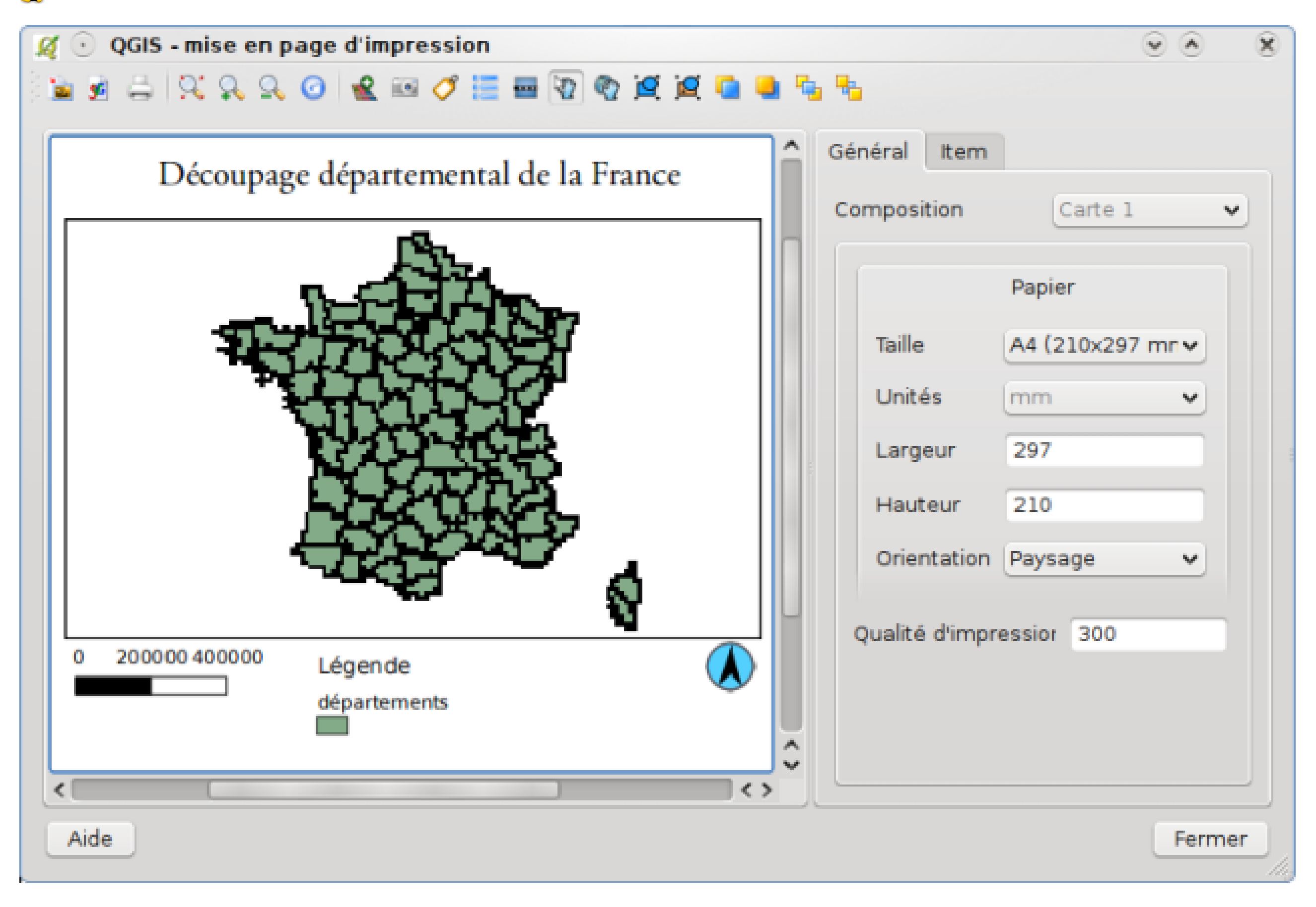


– Zoomer,
– Dézoomer,
– Zoom à l'étendue maximale et
– Rafraîchir la vue, si vous trouvez la vue dans un état incomplet.

10.1.4 Création de carte

La figure 33 montre le composeur de carte avec un exemple de mise en page incluant chaque type d'élément de la carte décrit dans la section au-dessus.

FIG. 33: Composeur de carte avec une vue de la carte, de la légende, de l'échelle graphique et du texte ajouté



Le composeur de carte vous permet de choisir plusieurs formats de sortie et il est possible de définir la résolution (qualité d'impression) et la taille du papier :

– L'icône Imprimer permet d'imprimer la mise en page à une imprimante ou dans un fichier PDF ou Postscript en fonction des pilotes d'imprimante installée.
– L'icône Exporter dans une image exporte le cadre du composeur dans plusieurs formats d'image tels que PNG, BPM, TIF, JPG, . . .
– L'icône Exporter au format SVG sauve le cadre du composeur de carte en SVG (Scalable Vector Graphic). **Note :** Actuellement la sortie SVG est très basique. Ce n'est pas un problème de

QGIS mais un problème de la bibliothèque Qt sous-jacente. Cela sera probablement corrigé dans une prochaine version.

11 Les extensions de QGIS

QGIS repose sur un système d'extensions. Ce dernier permet d'ajouter facilement de nouvelles fonctions au logiciel. De nombreuses nouvelles fonctions de QGIS sont implémentées comme des extensions **principales** ou **complémentaires**.

- Les **extensions principales** sont maintenues par l'équipe de développement de QGIS et sont intégrées automatiquement à chaque nouvelle distribution de QGIS. Elles sont écrites en C++ ou en Python. On trouvera plus d'informations sur les extensions principales dans la Section 12.
- Les **extensions complémentaires** sont actuellement toutes écrites en Python. Elles sont stockées dans des dépôts externes et maintenues par leurs auteurs. Elles peuvent être ajoutées à QGIS en utilisant l'extension principale nommée `Gestionnaire d'extensions`. On trouvera plus d'informations sur les extensions complémentaires dans la Section 13.

11.1 Gérer les extensions

De manière générale, gérer les extensions consiste à les afficher ou pas à l'aide du `Gestionnaire d'extension`. Les extensions complémentaires doivent d'abord être installées à l'aide du `Gestionnaire d'extension`.

11.1.1 Installer une extension principale

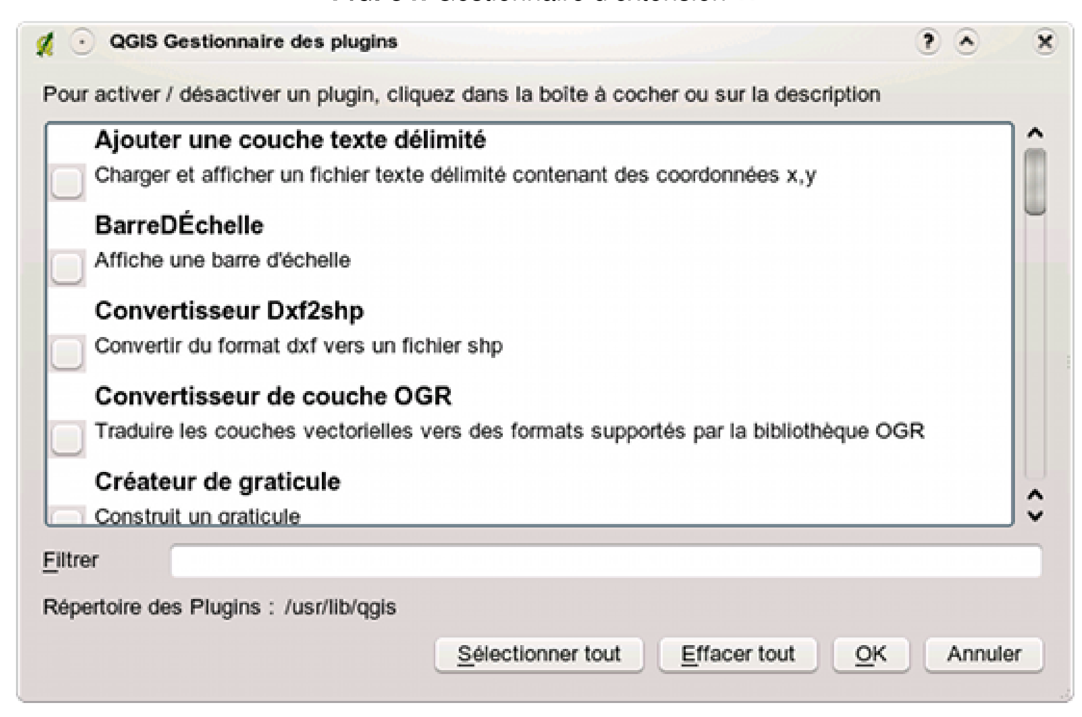
On installe une extension principale à l'aide du menu Plugins > Gestionnaire d'extension....

Le gestionnaire d'extension liste toutes les extensions disponibles et leur statut (installé ou pas). "Tous disponibles" signifie que toutes les extensions principales ou complémentaires que vous avez ajoutées à l'aide de l'extension `Gestionnaire d'Extension` (see Section 13). La figure 34 montre la boîte de dialogue du Gestionnaire d'extension. Les extensions installées sont mémorisées lorsque vous quittez l'application et seront restaurées à la prochaine ouverture de QGIS.

Astuce 40 EXTENSIONS ET PLANTAGES

Si votre installation de QGIS plante au démarrage, une extension est peut-être en cause. Vous pouvez éviter le chargement des extensions en éditant votre fichier de configuration (voir 4.7 pour localiser ce fichier). Localisez la configuration de l'extension et changez toutes les valeurs à "false" pour empêcher leur chargement. Par exemple, pour éviter le chargement de l'extension "Ajouter une couche texte délimitée", l'entrée de $HOME/.config/QuantumGIS/qgis.conf doit ressembler à `Add Delimited Text Layer=false`. Faites de même pour chaque extension dans la section [Extension]. Vous pouvez ensuite démarrer QGIS et ajouter les extensions une à la fois depuis le Gestionnaire d'extension pour déterminer celle qui est la source du problème.

FIG. 34: Gestionnaire d'extension



11.1.2 Installer une extension complémentaire de QGIS

Afin de pouvoir intégrer des extensions complémentaires dans QGIS, vous devez d'abord installer l'extension `Gestionnaire d'extension` comme décrit dans la Section 11.1.2. Les extensions python complémentaires peuvent ensuite être installées en deux étapes :

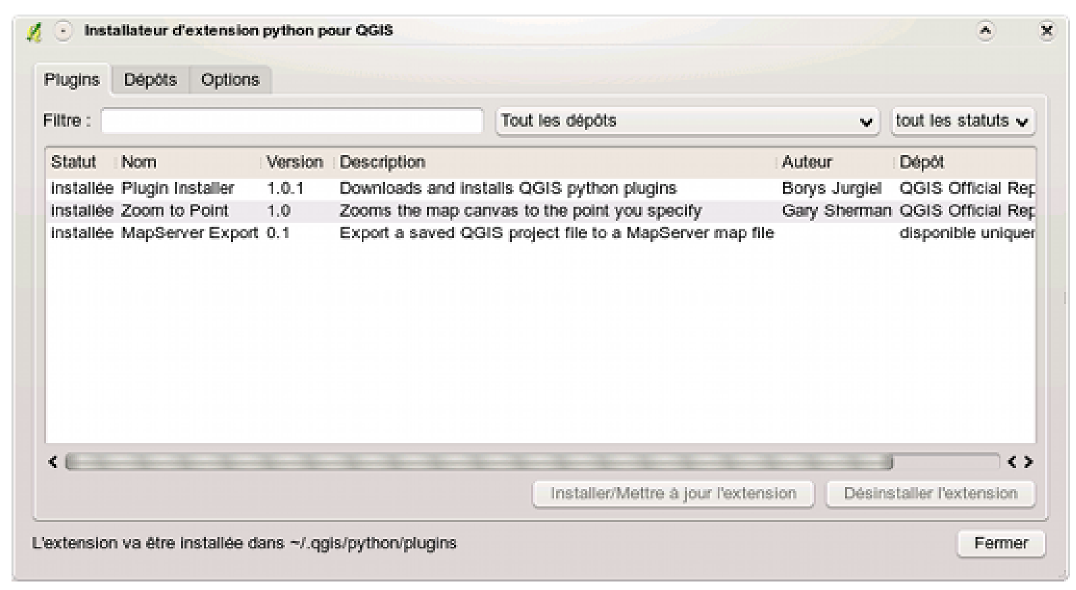
1. Télechargez une extension complémentaire depuis un dépôt à l'aide de l'`Installeur d'extension python` (Section 11.1.3). La nouvelle extension complémentaire sera intégrée dans la liste des extensions disponibles du `Gestionnaire d'extension`.
2. Installez l'extension à l'aide du `Gestionnaire d'extension`.

11.1.3 Utiliser l'installeur d'extension python de QGIS

Pour télécharger et installer une extension python complémentaire, cliquez sur le menu Plugins > Récupération des extensions python.... La fenêtre de l'`Installeur d'extension python` apparaîtra (figure 35) avec l'onglet Plugins, qui présente la liste de toutes les extensions python installées ou disponibles dans des dépôts distants. Chaque extension peut-être soit :

– **non installée** - signifie que l'extension est disponible dans le dépôt, mais n'est pas encore installée. Pour l'installer, sélectionnez-la dans la liste et cliquez sur le bouton Installer l'extension .

FIG. 35: Installer des extensions complémentaires python



– **nouveau** - même statut que ci-dessus, mais l'extension apparaît pour la première fois.
– **installée** - l'extension est installée. Si elle est également disponible dans un dépôt, le bouton [Ré-installer l'extension] est actif. En revanche, si la version disponible est plus ancienne que la version installée, le bouton [? ? ?] apparaît à la place.
– **mise à jour** - l'extension est installée, mais une version plus récente est disponible, le bouton [Mise à jour de l'extension] est actif.
– **invalide** - l'extension est installée, mais ne fonctionne pas. Les détails sont donnés dans la description de l'extension.

Onglet Plugins

Pour installer une extension, sélectionnez-la dans la liste et cliquez sur le bouton [Installer l'extension]. L'extension est installée dans son propre répertoire, par ex. pour dans `$HOME/.qgis/python/plugins` et n'est visible que pour l'utilisateur qui l'a installée. Voir une liste des répertoires d'installation des extensions sous d'autres OS dans la Section 15.3. Si l'installation réussit, un message de confirmation apparaît. Cliquez ensuite sur le menu [Plugins] > [Gestionnaire d'extension...] et chargez l'extension fraîchement installée. and load the installed plugin.

Si l'installation ne fonctionne pas, la raison est indiquée. Les problèmes les plus fréquents sont dus à des erreurs de connexion et à des modules Python manquants. Dans le premier cas, il vous faudra probablement attendre quelques minutes ou heures ; dans le second cas, il sera nécessaire d'installer les modules manquants avant d'utiliser les extensions. Pour Linux, les modules les plus recherchés devraient être disponibles dans un gestionnaire de paquets. Pour des installations

d'instruction dans Windows, visitez la page web du module. Si vous utilisez un proxy, vous pourrez avoir besoin de le configurer dans le menu Préférences > Options dans l'onglet Proxy .

Le bouton Désinstaller l'extension est actif seulement si l'extension sélectionnée est installée et n'est pas une extension principale. Notez que si vous avez installé une mise à jour d'une extension principale, il est toujours possible de désinstaller cette mise à jour avec le bouton Désinstaller l'extension et revenir à la version d'origine fournie à l'installation de Quantum GIS. Cette dernière ne peut pas être désinstallée.

Onglet Dépôts

Le second onglet Dépôts contient une liste de dépôts d'extensions disponibles. Par défaut, seul le dépôt officiel de QGIS (QGIS Official Repository) est utilisé. Vous pouvez ajouter des dépôts contenant des contributions d'utilisateurs, notamment le dépôt de contributions de QGIS (QGIS Contributed Repository) et quelques dépôts d'auteurs en cliquant sur le bouton Ajouter un dépôt-tiers d'extension à la liste . Ces dépôts contiennent une grande quantité d'extensions plus ou moins utiles, mais veuillez noter qu'elles ne sont pas maintenues par l'équipe de développement de QGIS et que nous ne pouvons prendre aucune responsabilité pour elles. Vous pouvez également gérer la liste de dépôts manuellement, pour ajouter, retirer ou éditer des entrées. Désactiver temporairement un dépôt particulier est possible en cliquant sur le bouton Editer .

La case X Chercher des mises à jour au démarrage configure QGIS pour rechercher des mises à jour et des actualités relatives aux extensions. Si la case est cochée, tous les dépôts listés et activés dans l'onglet Dépôts sont vérifiés à chaque démarrage du programme. Si une nouvelle extension ou une mise à jour pour une des extensions installées est disponible, une notification cliquable apparaît dans la barre de statut. Si la case est décochée, la recherche de mises à jour et d'actualités s'effectue uniquement lorsque l'Installeur d'Extension est lancé manuellement depuis le menu.

En cas de problèmes de connexion Internet, un indicateur *Recherche de nouvelles extensions...* dans la barre de statut peut rester visible durant toute la session QGIS et faire planter le programme à la fermeture. Dans ce cas, décochez la case.

11.2 Fournisseurs de données

Les Fournisseurs de données sont des extensions "spéciales" donnant accès à un dépôt de données. Par défaut, QGIS supporte les couches PostGIS et les bases de données fichiers couverts par la bibliothèque GDAL/OGR(Appendix A.1). L'utilisation d'extensions pour fournir des données permet d'élargir les sources de données utilisables par QGIS.

Les extensions fournissant des données sont automatiquement enregistrées par QGIS au démarrage. Elles ne sont pas gérées par le Gestionnaire d'Extension, mais utilisées en arrière-plan lors-

qu'un type de données est ajouté comme couche dans QGIS.

12 Utilisation des Extensions principales de QGIS

QGIS contient actuellement 17 extensions principales qui peuvent être chargées à l'aide du Gestionnaire d'extension. La table 6 présente la liste de chaque extension ainsi qu'une description de leur rôle et l'icône de barre d'outils. [9]

TAB. 6: Les extensions principales de QGIS

Icône	Extension	Description
	Ajouter une couche texte délimité	Charger et afficher des fichiers texte délimités contenant des coordonnées x et y
	Saisie de coordonnées	Saisir les coordonnées de la souris dans différents SRS
	Etiquette Copyright	Dessiner une étiquette copyright avec les informations associées
	Convertisseur DXF2SHP	Convertir un fichier Dxf en un fichier Shp
	Outils GPS	Charger et importer des données GPS
	GRASS	Activates the mighty GRASS Toolbox
	Géoréférencer	Ajouter des informations de projection aux fichiers raster
	Créateur de graticules	Créer un grid de latitude/longitude et l'enregistrer en tant que fichier Shp
	Interpolation	Effectuer des interpolations sur la bases des vertices d'une couche vecteur
	Export Mapserver	Exporter un projet QGIS en tant que mapfile Mapserver
	Flèche Nord	Afficher une flèche nord en superposition
	Convertisseur OGR	Convertir des couches vecteurs entre les formats pris en charge par OGR
	Installateur d'extensions	Télécharger et installer des extensions python pour QGIS
	SPIT	Outils d'import de fichiers Shp vers PostgreSQL/PostGIS
	Impression rapide	Imprimer rapidement une carte
	Barre d'échelle	Afficher une barre d'échelle en superposition
	WFS	Charger et afficher une couche WFS

[9] L'extension d'export Mapserver et l'installateur d'extension sont des extensions Python complémentaires, mais elles font partie du code source de QGIS et sont automatiquement chargées et sélectionnables dans le gestionnaire d'extension.

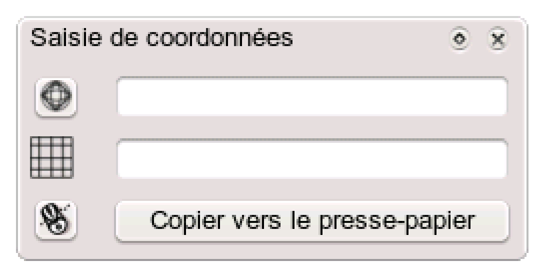
Astuce 41 SAUVEGARDER LA CONFIGURATION DES EXTENSIONS

Lorsqu'on enregistre un projet .qgs, tous les changements apportés aux extensions Flèche Nord, Barre d'échelle et Etiquette Copyright sont enregistrés, et sont restaurés à l'ouverture suivante du projet.

12.1 Extension de saisie de coordonnés

L'extension saisie de coordonnés, facile à utiliser, permet d'afficher des coordonnés sur la carte pour deux systèmes de coordonnés de référence (CRS) sélectionnés. Vous pouvez cliquer sur un point et copier les coordonnés dans le presse-papier, ou utiliser la fonction de suivi de la souris.

FIG. 36: L'extension Saisie de Coordonnés



1. Démarrez QGIS, sélectionnez Propriétés du Projet du menu Préférences et cliquez sur l'onglet Système de coordonnés de préférence (CRS) . Il est également possible de cliquer sur l'icône Statut de la projection, dans le coin bas droite de la barre de statut.
2. Cochez l'option X Autoriser la projection 'à la volée' et sélectionnez le système de coordonnés projeté "NAD27/Alaska Albers" dont l'EPSG est 2964 (voir aussi la Section 8).
3. Chargez la couche vecteur `alaska.shp` depuis le jeu de données échantillon de QGIS.
4. Chargez l'extension Saisie de coordonnés dans le Gestionnaire d'extensions (voir la Section 11.1.1) et cliquez sur l'icône Saisie de coordonnés. Une boîte de dialogue apparaît, comme indiqué dans la Figure 36.
5. Cliquez sur l'icône Cliquez ici pour sélectionner le SRS à utiliser pour l'affichage des coordonnés et sélectionnez le Système de coordonnés Géographiques `WGS84` (EPSG 4326).
6. Vous pouvez désormais cliquer n'importe où dans la fenêtre carte et les coordonnés des points sélectionnés s'afficheront en "NAD27/Alaska Albers" et `WGS84`, comme indiqué dans la Figure 36.
7. Pour activer le suivi des coordonnés du curseur de la souris, cliquez sur l'icône  suivi du curseur.
8. Vous pouvez également copier les coordonnés vers le presse-papiers.

12.2 Extensions Décorations

Les extensions Décorations incluent les extensions Etiquette Copyright, Flèche Nord et Echelle Graphique. Elles sont destinées à "habiller" la carte en ajoutant des éléments cartographiques.

12.2.1 l'extension Etiquette Copyright

Le nom de cette extension est sujet à confusion : elle permet d'ajouter n'importe quelle étiquette de texte à la carte.

FIG. 37: l'extension Etiquette Copyright

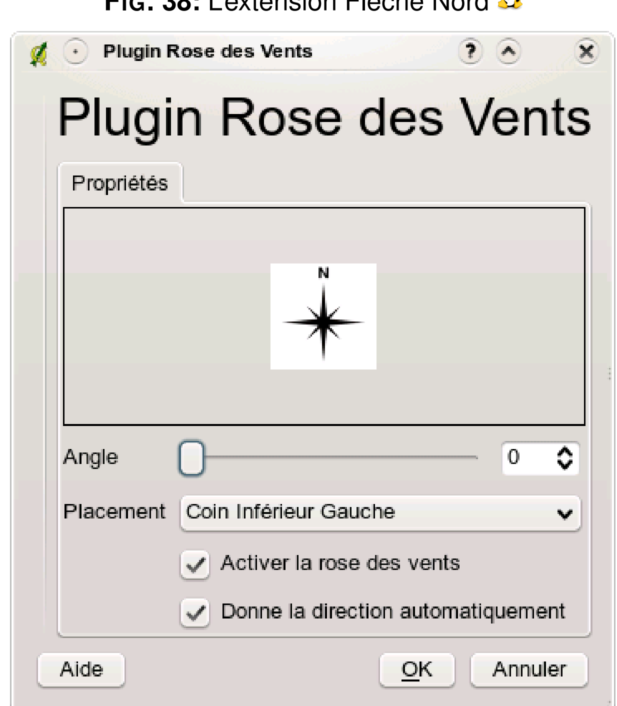
1. Assurez-vous que l'extension soit chargée.
2. Cliquez sur Plugins > Décorations > © Etiquette de Copyright ou utilisez le bouton Etiquette de Copyright dans la barre d'outils.
3. Entrez le texte que vous souhaitez placer sur la carte. Vous pouvez utiliser le HTML, comme indiqué dans l'exemple.
4. Choisissez la position de l'étiquette depuis la liste de choix déroulant Placement Coin Inférieur Droit.
5. Assurez-vous que la case [X] Activer l'étiquette du copyright soit cochée.
6. Cliquez sur OK .

Dans l'exemple ci-dessus, la première ligne est en gras, la seconde (créée avec
) contient un symbole copyright, suivi par le nom de notre société en italique.

12.2.2 L'extension Flèche Nord

L'extension Flèche Nord place une simple rose des vents sur la carte. Un seul style est disponible pour le moment. Vous pouvez ajuster l'angle de la flèche ou laisser QGIS déterminer automatiquement la direction. Si vous faites ce dernier choix, QGIS essaiera de calculer la meilleure orientation de la flèche. Quatre options sont disponibles pour la position de la flèche, qui correspondent au quatre coins de la fenêtre carte.

FIG. 38: L'extension Flèche Nord

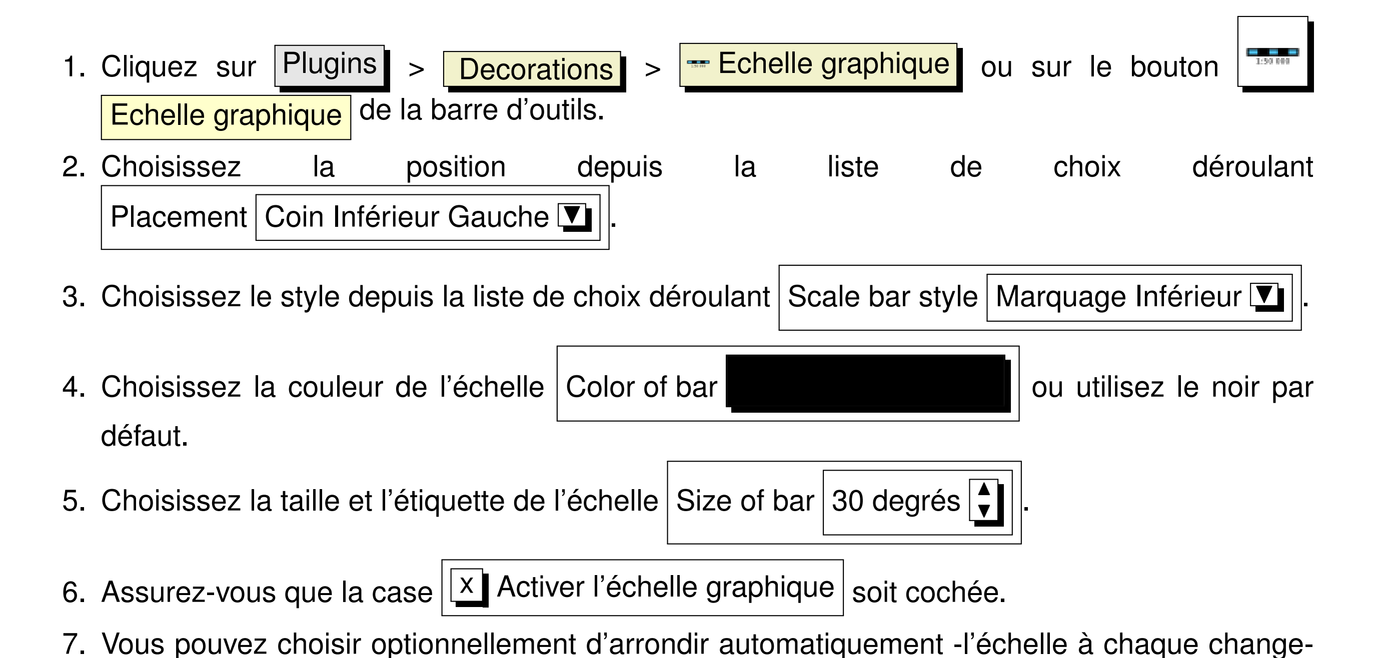


12.2.3 L'extension Échelle Graphique

L'extension Échelle Graphique ajoute une simple échelle graphique à la carte. Vous pouvez contrôler le style et la position, ainsi que l'étiquetage de l'échelle.

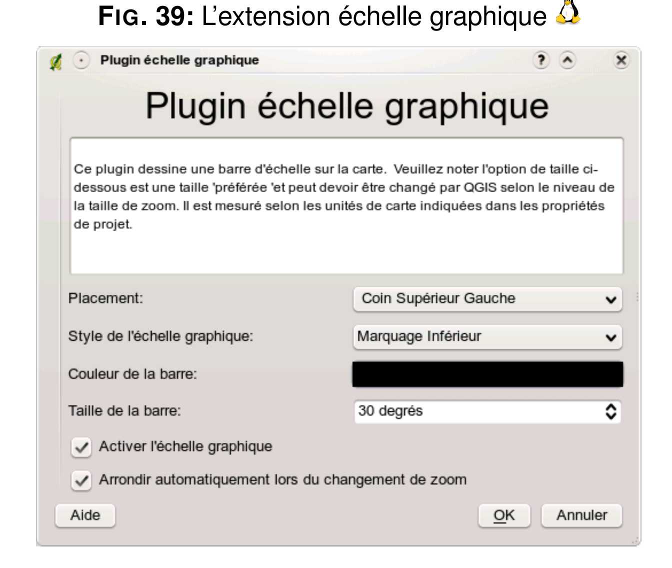
L'échelle ne peut afficher que les mêmes unités que celles de votre fenêtre carte. Si les unités des couches sont en mètres, vous ne pourrez pas créer une échelle en pieds. De la même manière, si vous utilisez les degrés décimaux, vous ne pourrez pas créer une échelle en mètres.

Pour ajouter une échelle graphique :



1. Cliquez sur Plugins > Decorations > Echelle graphique ou sur le bouton Echelle graphique de la barre d'outils.
2. Choisissez la position depuis la liste de choix déroulant Placement Coin Inférieur Gauche.
3. Choisissez le style depuis la liste de choix déroulant Scale bar style Marquage Inférieur.
4. Choisissez la couleur de l'échelle Color of bar ou utilisez le noir par défaut.
5. Choisissez la taille et l'étiquette de l'échelle Size of bar 30 degrés.
6. Assurez-vous que la case X Activer l'échelle graphique soit cochée.
7. Vous pouvez choisir optionnellement d'arrondir automatiquement -l'échelle à chaque changement de niveau de zoom X Arrondir automatiquement lors du changement de zoom
8. Cliquez sur OK .

FIG. 39: L'extension échelle graphique



12.3 Extension de texte Délimité

L'extension Fichier texte délimité permet de charger un fichier texte délimité comme couche dans QGIS.

Exigences

Pour afficher un fichier texte délimité comme couche, le fichier texte doit contenir :

1. Une ligne d'entête délimitée avec les noms de champs. Cette ligne doit être la première du fichier texte.
2. La ligne d'entête doit contenir des champs X et Y. Ces champs peuvent avoir n'importe quel nom.
3. Les coordonnées X et Y doivent être de type numérique. Le système de coordonnées n'est pas important.

Comme exemple de fichier texte valide, nous pouvons importer le fichier de points d'élévation `elevp.csv` fourni avec le jeu de données échantillon de QGIS (Voir Section 3.2) :

```
X;Y;ELEV
-300120;7689960;13
-654360;7562040;52
1640;7512840;3
[...]
```

On notera les points suivants à propos du fichier texte :

1. Le fichier texte d'exemple utilise ; comme délimiteur. N'importe quel caractère peut être utilisé pour délimiter les champs.
2. La première ligne est la ligne d'entête. Elle contient les champs X, Y et ELEV.
3. Les guillemets (") ne peuvent pas être utilisés pour délimiter les champs de texte.
4. les coordonnées x sont inclues dans le champ *X*.
5. les coordonnées y sont inclues dans le champ *Y*.

Utilisation de l'extension

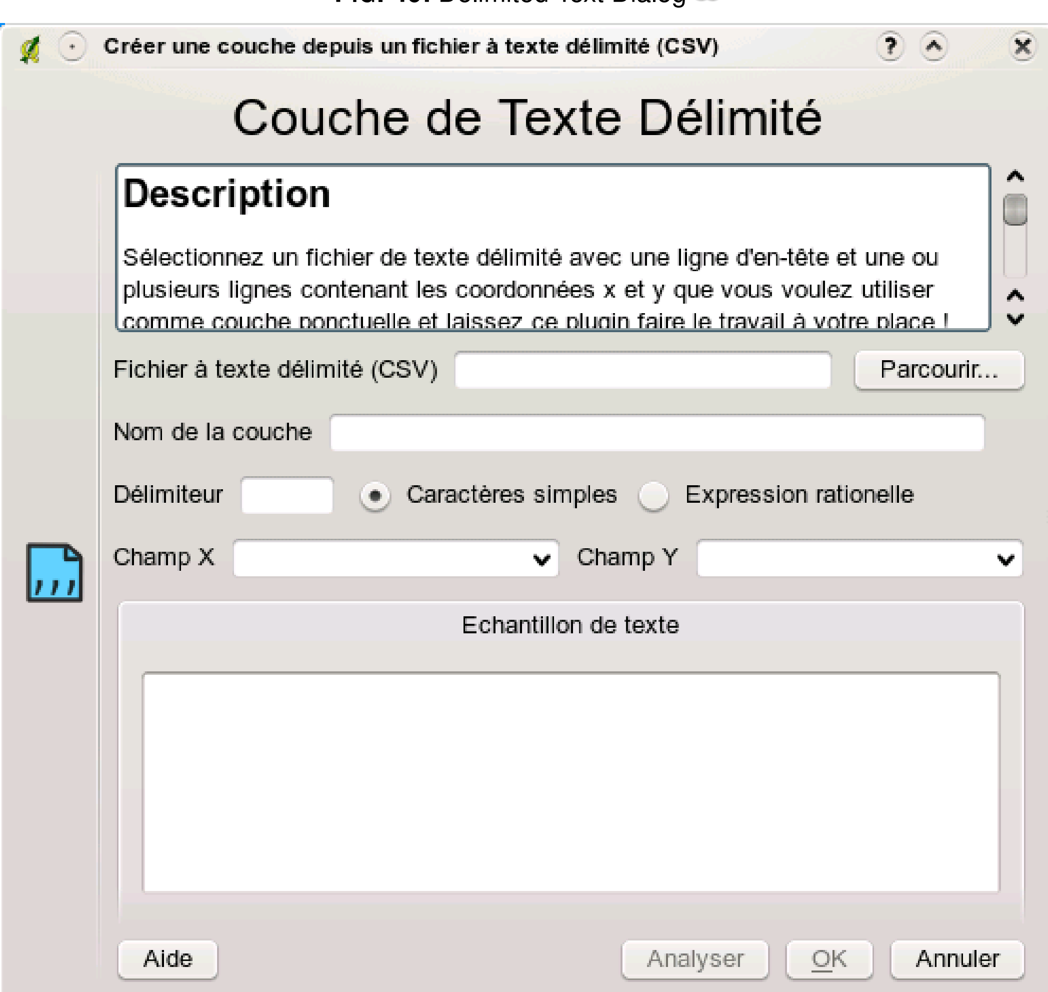
Pour utiliser l'extension, QGIS doit être lancé. Utilisez le Gestionnaire d'extensions pour charger l'extension :

Démarrez QGIS, puis ouvrez le gestionnaire d'extensions en cliquant sur Extensions > Gestionnaire d'extensions...

Le gestionnaire d'extension affiche une liste des extensions disponibles. Celles qui sont déjà chargées sont cochées en début de ligne. Cochez la case de l'extension [X] Add Delimited Text Layer et cliquez sur OK pour charger l'extension comme indiqué dans la Section 11.1.

Cliquez sur la nouvelle icône de la barre d'outils Ajoutez un fichier texte délimité pour ouvrir la boîte de dialogue Texte Délimité comme indiqué dans la Figure 40.

FIG. 40: Delimited Text Dialog



Sélectionnez d'abord le fichier `qgis_sample_data/csv/elevp.csv` à importer en cliquant sur le bouton Browse . Une fois que le fichier est sélectionné, l'extension va tenter d'analyser le fichier en utilisant le dernier délimiteur utilisé, en l'occurrence ;. Pour analyser correctement le fichier, il est important de sélectionner le bon délimiteur. Pour changer le délimiteur en tab, utilisez \t (expression habituelle du caractère tab). Après avoir changé le délimiteur, Cliquez sur analyser .

Choisissez les champs X et Y depuis les listes déroulantes et entrez un nom de couche (par exemple `elevp`) comme indiqué dans la Figure 40. Pour ajouter la couche à la carte, cliquez sur [Add Layer]. Le fichier texte délimité se comporte maintenant comme n'importe quel autre calque dans QGIS.

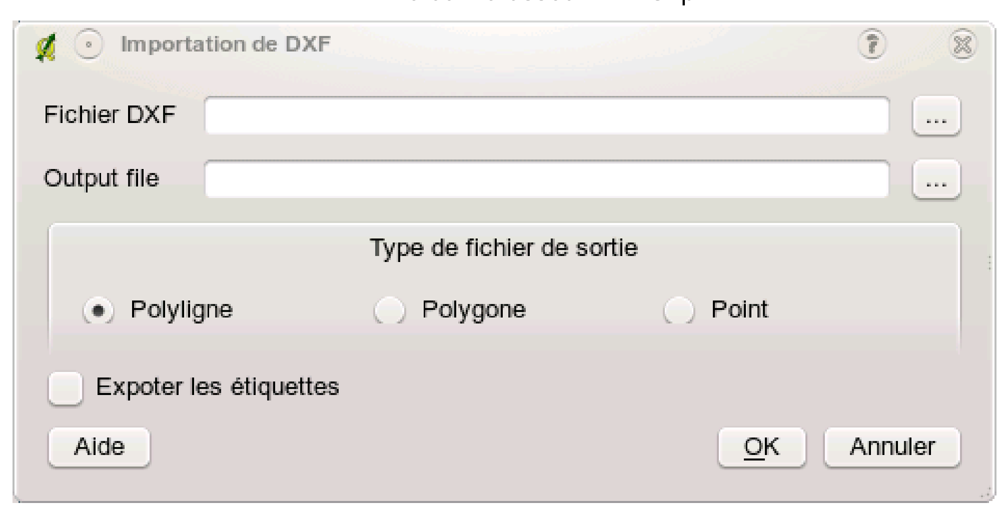
12.4 Dxf2Shp Converter Plugin

12.5 Extension de conversion Dxf2Shp

L'extension de conversion Dxf2Shp permet de convertir des données vectorielles du format DXF au format shapefile (SHP). Très facile à utiliser, elle présente les fonctionnalités présentées dans la Figure 41).

- **Fichier DXF** : Entrez l'adresse du fichier DXF à convertir.
- **Fichier SHP de sortie** : Entrez le nom souhaité du fichier shape à créer.
- **Type de fichier de sortie** : Spécifiez le type du fichier shape. Les formats implémentés pour le moment sont polyligne, polygone et point.
- **Exporter les étiquettes** : Si vous cochez cette case, une couche supplémentaire sera créée (points), et la table dbf associée contiendra des informations à propos des champs "TEXT" trouvés dans le fichier DXF, et les chaînes de caractères elles-mêmes.

FIG. 41: Le convertisseur Dxf2Shp



1. Démarrez QGIS, chargez l'extension Dxf2Shp dans le gestionnaire d'extensions (voir la Section 11.1.1) et cliquez sur l'icône Convertisseur Dxf2Shp qui apparait dans les barres d'outils de QGIS. La boîte de dialogue de l'extension Dxf2Shp apparaît, comme indiqué dans la Figure 41.
2. Entrez le fichier d'origine DXF, choisissez un nom pour le fichier SHP de sortie et le type de fichier shape.
3. Cochez la case [X] Exporter les étiquettes, si vous souhaitez créer une couche supplémentaire comprenant les étiquettes.

4. Cliquez sur [Ok].

12.6 L'extension Géoréférencer

Le Géoréférencer permet de générer un fichier world file pour les rasters (ce fichier contient toutes les informations nécessaires au positionnement spatial). Il faut pour cela, sélectionner des points du raster et leur associer des coordonnées ; l'extension calculera les paramètres du fichier world file. Plus grand sera le nombre de coordonnées fournies, meilleur sera le résultat.

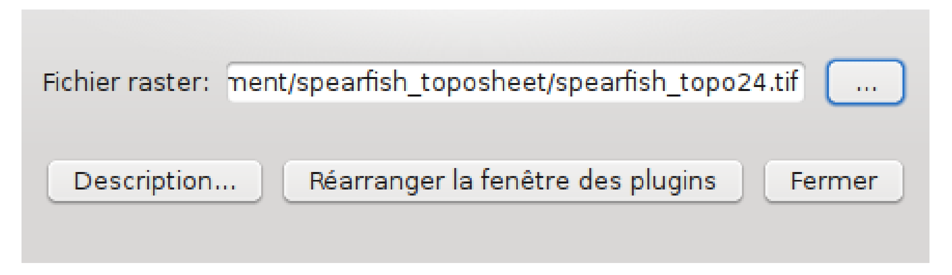
En guise d'exemple, nous générerons un fichier world file pour une carte topographique du Dakota du Sud publiée par le SDGS. Elle pourra, par la suite, être affichée avec les données du secteur GRASS spearfish60. La carte topographique peut être téléchargée à l'adresse suivante : `http://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz`

La première étape est le téléchargement de ce fichier et son extraction.

```
wget http://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz
tar xvzf spearfish_toposheet.tar.gz
cd spearfish_toposheet
```

L'étape suivante consiste à démarrer QGIS, charger l'extension Géoréférencer et sélectionner le fichier `spearfish_topo24.tif`.

FIG. 42: Sélectionner une image à géoréférencer

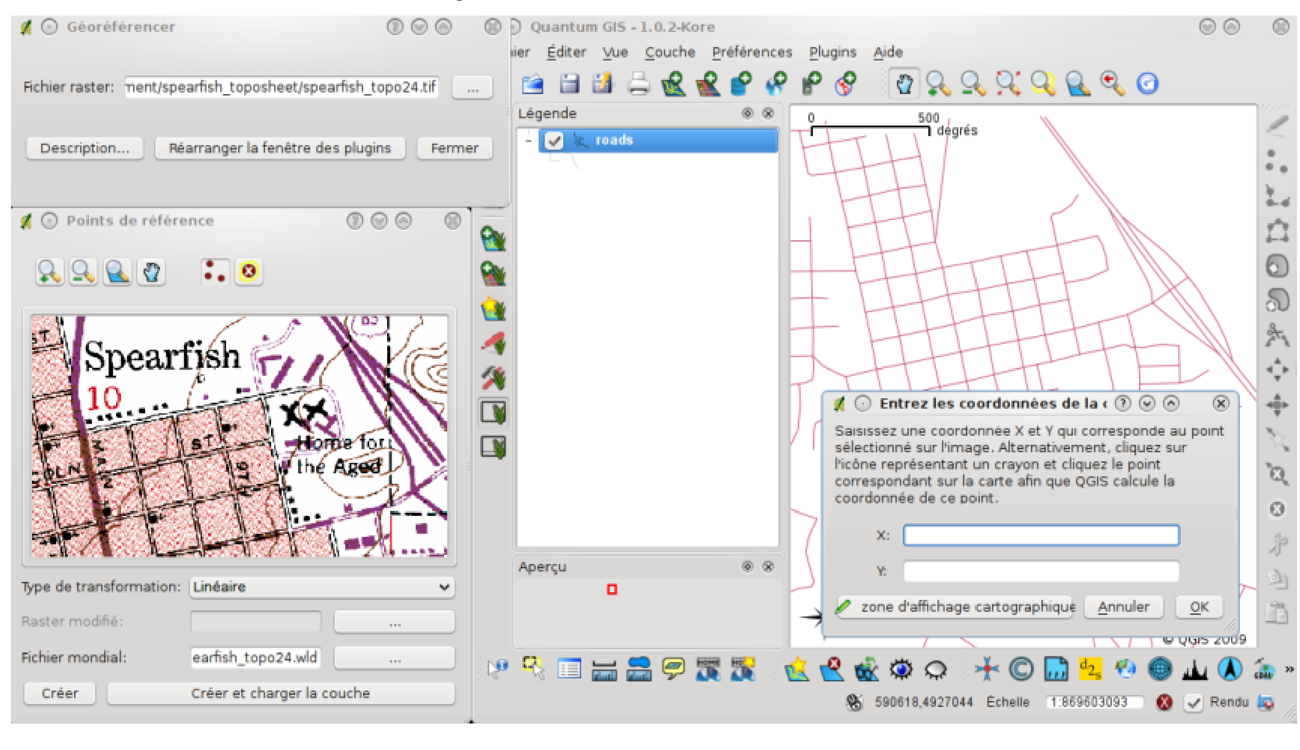


Cliquer ensuite sur le bouton Arrange plugin window pour ouvrir l'image dans la fenêtre de géoréférencement et pour organiser les différentes fenêtres de QGIS sur le bureau (voir Figure 43).

Avec le bouton Add Point on peut commencer à ajouter des points sur l'image raster et entrer leurs coordonnées. L'extension calculera les paramètres du fichier world file (voir Figure 44). Plus on fournit de coordonnées, meilleur sera le résultat. Il y a deux manières de procéder :

1. Cliquer en un point de la carte raster et entrer les coordonnées X et Y manuellement
2. Cliquer en un point de la carte raster puis sur le bouton from map canvas pour ajouter les coordonnées X et Y à l'aide d'une carte géoréférencée déjà chargée dans QGIS.

FIG. 43: Organiser les fenêtres de QGIS sur le bureau



Pour cet exemple, nous utilisons la seconde procédure et entrons les coordonnées des points sélectionnés à l'aide de la carte `roads` fournie avec le secteur `spearfish60` disponible à l'adresse suivante : `http://grass.osgeo.org/sampledata/spearfish_grass60data-0.3.tar.gz`

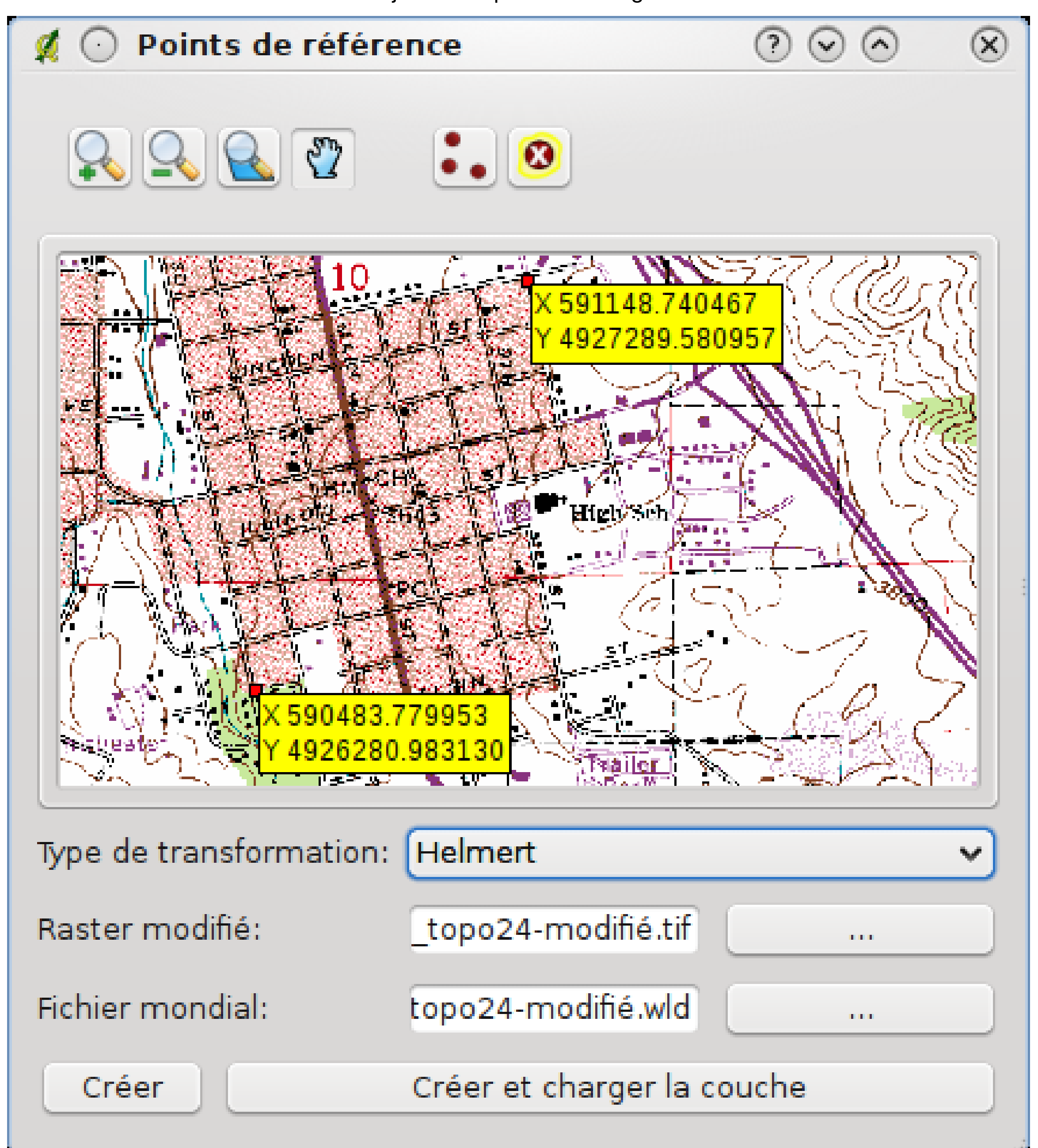
Si vous ne savez pas comment intégrer le secteur `spearfish60` avec l'extension GRASS, des explications sont disponibles dans la section 9. Comme on peut le voir dans la Figure 44, l'outil de géoréférencement dispose de boutons pour le changement d'échelle, la translation, l'ajout et la suppression de points dans l'image.

Après avoir ajouter suffisamment de points à l'image, on peut sélectionner le type de transformation pour le processus de géoréférencement et sauver le fichier world file avec le fichier Tiff. Dans notre exemple, nous avons choisi [Transform type | linear transformation ▼] bien que [Transform type | Helmert transformation ▼] pourrait tout aussi bien convenir.

Astuce 42 CHOISIR LE TYPE DE TRANSFORMATION

La transformation linéaire (affine) est une tranformation de 1er ordre et est utilisée pour la mise à l'échelle, la translation et la rotation d'images géométriquement correctes. Avec la transfomation de Helmert, on ajoute simplement des informations de type géocodage à l'image. Si l'image est distordue, il sera nécessaire d'utiliser une logiciel permettant les transformations polynomiales de 2nd et 3ème ordre, comme GRASS GIS par exemple

FIG. 44: Ajouter les points à l'image raster

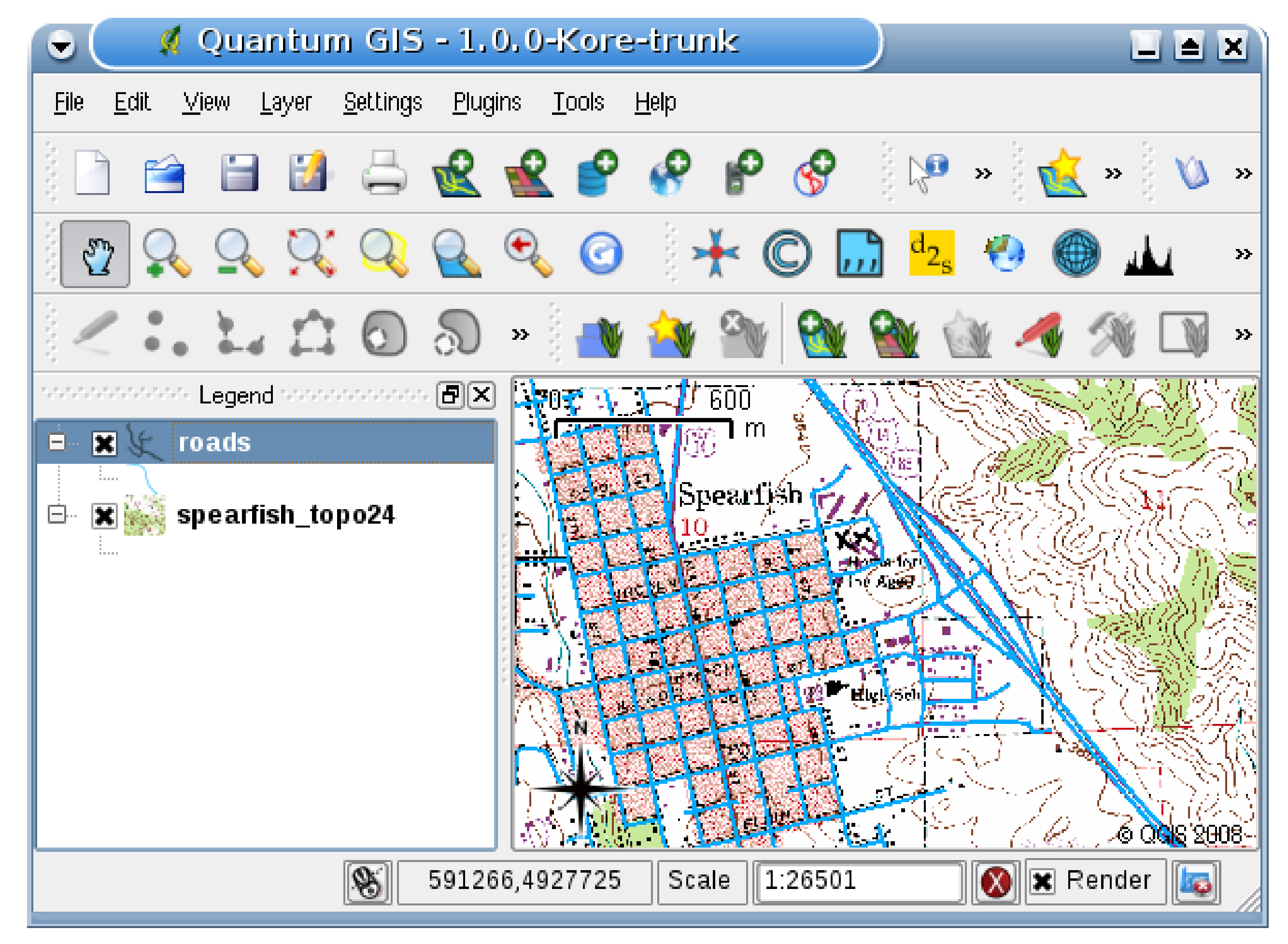


Les points ajoutés à la carte seront conservés dans un fichier `spearfish_topo24.tif.points` adjoint à l'image raster. Ceci permet d'optimiser le résultat en ouvrant de nouveau l'extension Géoréférencer et en ajoutant de nouveaux points ou supprimant des points existants. Le fichier `spearfish_topo24.tif.points` de cet exemple contient les points suivants :

```
mapX      mapY      pixelX  pixelY
591630.196867999969982  4927104.309682800434530 591647   4.9271e+06
608453.589164100005291  4924878.995150799863040 608458   4.92487e+06
602554.903929700027220  4915579.220743400044739 602549   4.91556e+06
591511.138448899961077  4915952.302661700174212 591563   4.91593e+06
602649.526155399973504  4919088.353569299913943 602618   4.91907e+06
```

Cinq points ont été utilisés pour géoréférencer l'image raster. Pour obtenir des résultats corrects, il est important de répartir les points régulièrement sur l'image. Finalement on peut vérifier le résultat, charger la nouvelle carte géoréférencée `spearfish_topo24.tif` et la superposer à la carte `roads` du secteur `spearfish60`.

FIG. 45: Carte géoréférencée avec superposition des routes provenant du secteur spearfish60



12.7 Extension Impression Rapide

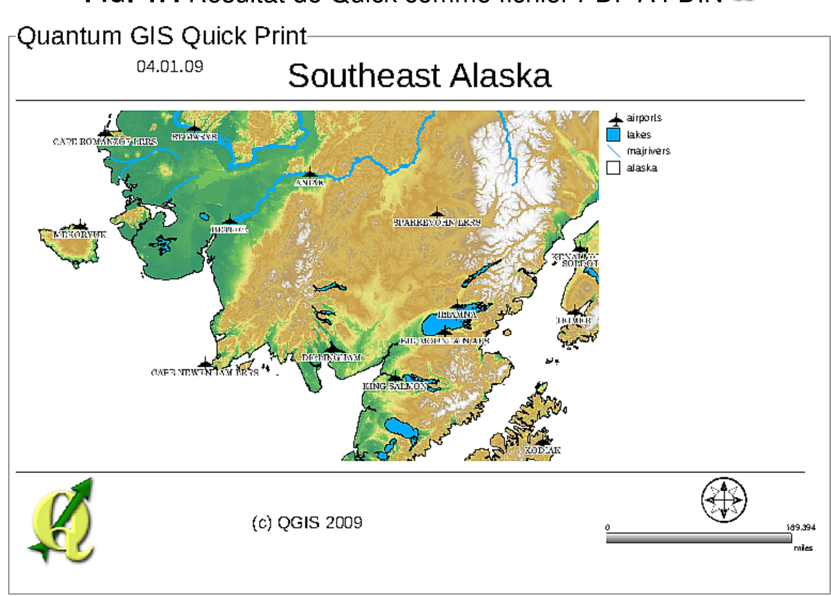
L'extension Impression Rapide permet d'imprimer le cadre de la carte actuelle avec un minimum d'effort au format PDF. Tout ce qu'a besoin d'ajouter l'utilisateur est un titre pour la carte, un nom pour la carte et la taille de la page (voir Figure 46)

FIG. 46: Boîte de dialogue de l'Impression Rapide



La figure 47 ci-dessous montre le résultat de l'Impression Rapide dans un fichier A4 DIN à partir du jeu de données d'échantillon de l'Alaska. Si vous désirez plus de contrôle sur la mise en page de la carte, utiliser plutôt l'extension composeur de carte, décrit dans la section 10.

FIG. 47: Résultat de Quick comme fichier PDF A4 DIN



12.8 Extension GPS

12.8.1 Qu'est ce qu'un GPS ?

Le GPS, Global Positioning System, est un système basé sur des satellites qui permet à toute personne possédant un récepteur GPS d'obtenir sa position exacte n'importe où dans le monde. Il est utilisé comme aide à la navigation, comme par exemple pour les avions, dans les bateaux et par les voyageurs. Le récepteur GPS utilise les signaux des satellites pour calculer la latitude, la longitude et (parfois) l'élévation. La plupart des récepteurs ont également la possibilité de stocker la position (nommé *waypoints*), des séquences de positions qui constituent un *itinéraire* prévu et un tracklog ou *track* des déplacements du récepteur en fonction du temps. Waypoints, itinéraires et tracks sont les trois types d'objet basic dans les données GPS. QGIs affiche les waypoints dans des couches points tandis que les itinéraires et les tracks sont affichés dans des couches linéaires.

12.8.2 Charger des données GPS à partir d'un fichier

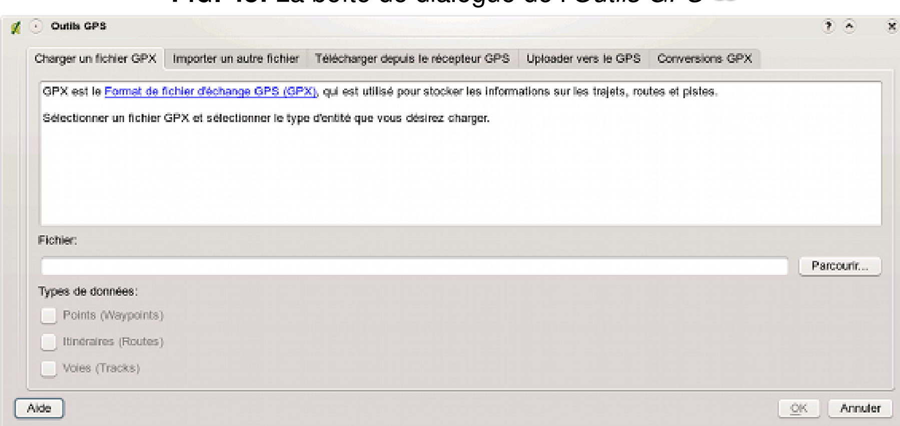
Il y a des dizaines de formats de fichier différent pour stocker des données GPS. Le format que QGIS utilise est appelé GPX (GPS eXchange format), qui est un format d'échange standard qui peut contenir n'importe quel nombre de waypoints, itinéraire et tracks dans un même fichier.

Pour charger un fichier GPX vous devez d'abord charger le plugin Plugins > Gestionnaire de Plugin > x Outils GPS. Quand cette extension est chargée, un bouton avec un petit périphérique GPS apparaîtra dans la barre d'outils. Un fichier GPX d'exemple est disponible dans le jeu de données d'exemple de QGIS : `/qgis_sample_data/gps/national_monuments.gpx`. Voir Section 3.2 pour plus d'information sur les données d'exemple.

1. cliquez sur l'icône Outils GPS et ouvrez l'onglet Charger un fichier GPX
2. Naviguez vers le répertoire `qgis_sample_data/gps/`, sélectionnez le fichier GPX `national_monuments.gpx` et cliquez sur le bouton Ouvrir .

Utilisez le bouton ... pour sélectionner le fichier GPX, puis utilisez la case à cocher pour sélectionner les types de géométrie que vous voulez charger à partir de ce fichier GPX. Chaque type d'objet sera chargé dans une couche séparée lors du clique sur le bouton OK . Le fichier `national_monuments.gpx` inclue seulement des waypoints.

FIG. 48: La boîte de dialogue de l'*Outils GPS*



12.8.3 GPSBabel

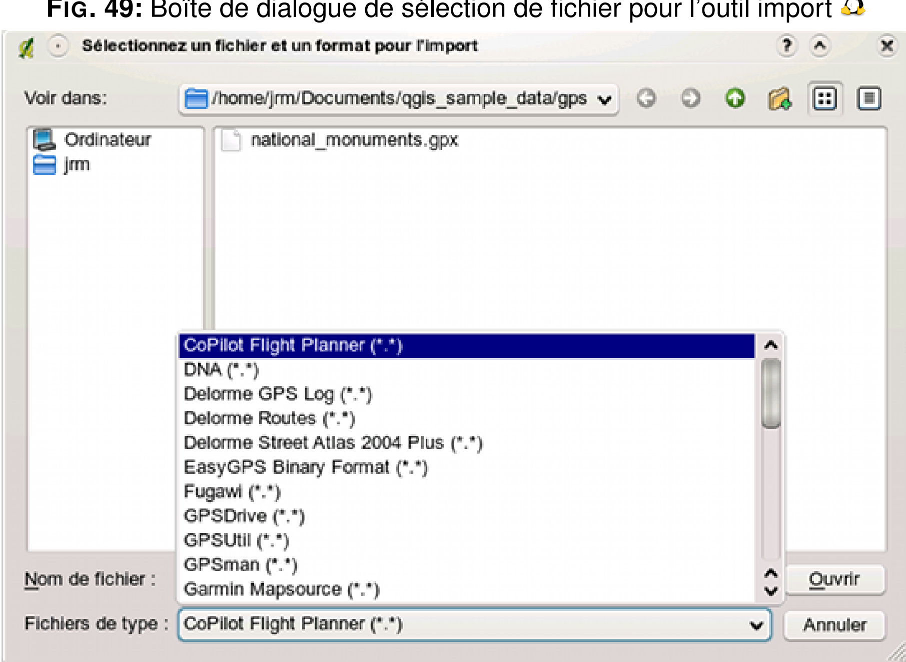
Puisque QGIs utilise des fichiers GPX vous avez besoin de convertir les autres formats de fichiers GPS en GPX. Cela peut être réalisé pour plusieurs formats en utilisant le logiciel libre GPSBabel qui est disponible sur `http://www.gpsbabel.org`. Ce programme peut aussi transférer des données GPS entre votre ordinateur et un périphérique GPS. QGIS utilise GPSBabel pour réaliser ces tâches, il est donc recommandé de l'installer. Cependant si vous voulez juste charger des données à partir de fichiers GPX vous n'en avez pa besoin. La version 1.2.3 de GPSBabel est connue pour bien fonctionner avec QGIS, mais vous pouvez devriez pouvoir utiliser des versions plus récentes sans problème.

12.8.4 Importer des données GPS

Pour importer des données d'un fichier qui n'est pas un fichier GPX, vous devez utiliser l'outil [Importer d'autres fichiers] dans la boîte de dialogue des outils GPS. Vous sélectionnez le fichier que vous voulez importer, quel type de géométrie vous voulez importer de ce celui-ci, où vous voulez stocker le fichier GPX converti et sous quel nom enregistrer la couche.

Quand vous sélectionnez le fichier à importer vous devez également sélectionner le format de ce fichier en utilisant le menu dans la boîte de dialogue de sélection de fichiers (voir la figure 49). Tous les formats ne gèrent pas les 3 types de géométries, vous pourrez donc choisir pour plusieurs formats qu'un ou deux types.

FIG. 49: Boîte de dialogue de sélection de fichier pour l'outil import



12.8.5 Télécharger des données GPS à partir d'un périphérique

QGIS peut utiliser GPSBabel pour télécharger des données d'un périphérique GPS directement dans des couches vecteurs. Pour cela, utilisez l'outil Télécharger d'un GPS (voyez la figure 50), où vous choisissez votr etype de périphérique GPS, le port auquel il est connecté, le type de géométrie que vous voulez télécharger, le fichier GPX où les données doivent être stockées et le nom de la nouvelle couche.

FIG. 50: L'outil de téléchargement

Le type de périphérique que vous sélectionner dans le menu périphérique GPS détermine comme GPSBabel tente de communiquer avec votre périphérique. Si aucun des types ne fonctionne avec votre périphérique GPS, vous pouvez créer un nouveau type adapté (voir la section 12.8.7).

Le port est un nom de fichier ou autre nom que votre système d'opération utilise comme une référence du port physique de votre ordinateur auquel le périphérique GPS est connecté. Sous Linux cela ressemble à /dev/ttyS0 ou /dev/ttyS1 et sous Windows c'est COM1 ou COM2.

Quand vous cliquez sur le bouton OK les données seront téléchargées du périphérique et apparaîtront dans une couche dans QGIS.

12.8.6 Envoyer des données GPS vers un appareil

Vous pouvez également envoyer directement vos données d'une couche vecteur dans QGIS vers un périphérique GPS en utilisant l'outil Upload to GPS . La couche doit être une couche GPX. Pour réaliser cela, vous sélectionnez simplement la couche que vous voulez uploader, le type de votre périphérique GPS et le port auquel il est connecté. De la même manière que pour l'outil de téléchargement, vous pouvez définir de nouveaux types de périphérique si le vôtre n'est pas dans la liste.

Cet outil est très utile avec les possibilités d'édition vectorielle de QGIS. Vous pouvez charger une carte, créer des waypoints et des routes puis les télécharger et les utiliser dans votre périphérique QGPS.

12.8.7 Définir de nouveaux types de périphériques

Il y a beaucoup de types différents de périphériques GPS. Les développeurs de QGIS ne peuvent pas les tester tous, si vous en avez un qui ne fonnctione pas avec un des types de périphériques dans les outils Récupérer du GPS et Télécharger du GPS , vous pouvez définir votre propre type de périphérique. Vous réalisez cela en utilisant l'éditeur de périphérique de GPS, que vous démarrez en cliquant le bouton Éditer un périphérique dans la fenêtre de récupération ou de téléchargement.

Pour définir un nouveau périphérique, vous cliquez sur le boutton Nouveau périphérique , entrez un nom, une commande de récupération et une commande de téléchargement pour votre périphérique, et cliquez sur le bouton Mise à jour du périphérique . Le nom sera listé dans les menus des périphériques dans la fenêtre récupération et téléchargement, et peut être n'importe quelle chaîne de caractère. La commande de récupération est la commande qui est utilisée pour récupérer les données du périphérique vers un fichier GPX. Ce sera certainement une commande GPSBabel, mais vous pouvez utiliser un autre programme en ligne de commande qui créé un fichier GPX. QGIS remplacera les mots-clé `%type`, `%in`, et `%out` lorsqu'il lancera la commande.

%type sera remplacé par "-w" si vous téléchargez des waypoints, "-r" si vous téléchargez des routes et "-t" si vous téléchargez des tracks. Ce sont des options de la ligne de commande qui dit à GPS-Babel quel type d'objet à télécharger.

%in sera remplacé par le port que vous avez choisi dans la boîte de dialogue de téléchargement et %out sera remplacé par le nom que vous avez choisi pour le fichier GPX où les données téléchargées doivent être stockées. Si vous créez un type de périphérique avec la commande de téléchargement "gpsbabel%type -i garmin -o gpx %in %out" (c'est actuellement la commande de téléchargement pour le type de périphérique prédéfini GPS device : Garmin serial ▼) puis utilisez-le pour télécharger les waypoints à partir du port "/dev/ttyS0" vers le fichier "output.gpx", QGIS remplacera les mots-clés et lancera la commande "gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx".

La commande de téléchargement est la commande qui est utilisée pour télécharger des données vers le périphérique. Le même mot-clé est utilisé, mais %in est maintenant remplacé par le nom du fichier GPX pour la couche qui est en téléchargée et %out est remplacé par le nom du port.

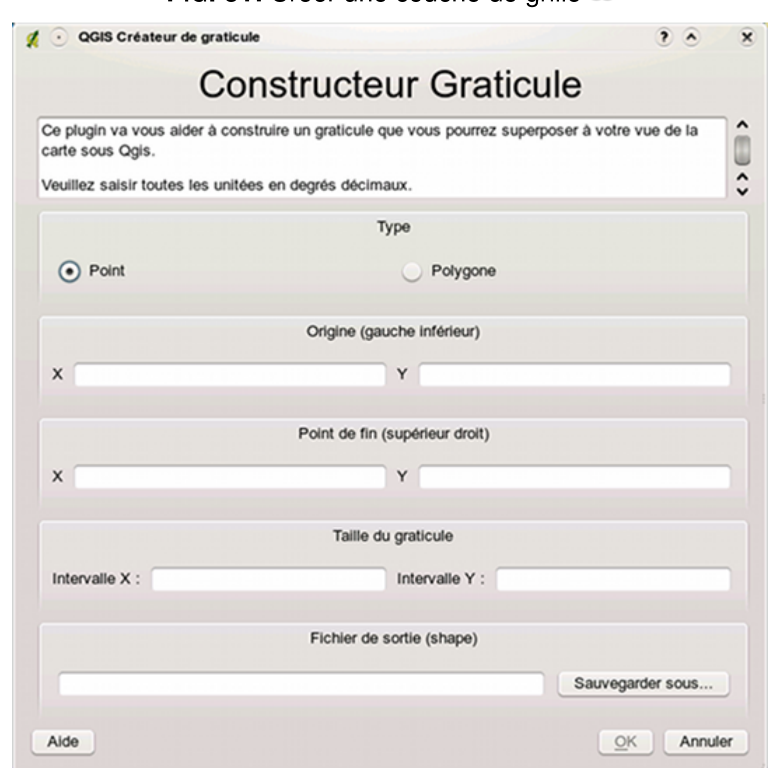
Vous pouvez en savoir plus sur GPSBabel et ses options de ligne de commande disponible sur http://www.gpsbabel.org.

Une fois le nouveau type de périphérique créé celui-ci apparaitra dans les listes de périphérique dans les outils de récupération et de téléchargement.

12.9 Extension Créateur de grille

Le créateur de grille permet de créer une “grille” de points ou de polygones pour couvrir notre zone d’intérêt. Toutes les unités doivent être entrées en degrés décimaux. Le format de sortie est le shapefile qui peut être projeté à la volée pour correspondre à vos données.

FIG. 51: Créer une couche de grille



Voici un exemple pour créer une grille :

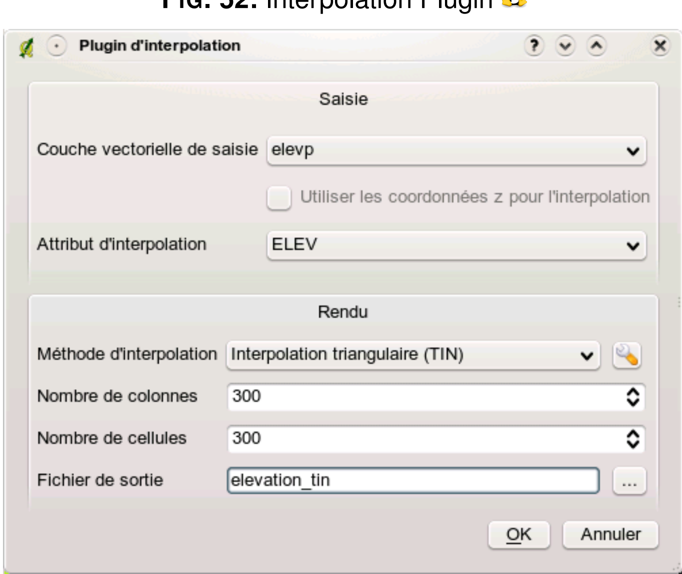
1. Démarrez QGIS, charger l’extension "Création de grille" dans le Gestionnaire de plugin (voir la section 11.1.1) et cliquez sur l’icône Créateur de grille qui apparait dans la barre de menu QGIS.
2. Choisissez le type de grille que vous voulez créer : point ou polygone.
3. Entrez la latitude et la longitude pour les coins bas-gauche et haut-droit de la grille.
4. Entrez l’intervalle à utilisé dans la construction de la grille. Vous pouvez entrer différentes valeurs pour les directions X et Y (longitude, latitude).
5. Choisissez le nom et le répertoire du shapefile à créé.
6. cliquez sur OK pour créer la grille et l’ajouter à la carte.

12.10 Extension d'interpolation

Cette extension permet d'interpoler une couche raster TIN ou IDW depuis une couche vecteur de points, elle est très facile à utiliser comme le montre la figure 52.

– **Couche vectorielle de saisie** : Sélectionnez une couche vectorielle de points présente dans la fenêtre carte de QGIS.
– **Interpolation attribute** : Sélectionnez la colonne attributaire utilisée pour l'interpolation, ou cochez la case [X] Utilisez les coordonnées Z pour l'interpolation.
– **Méthode d'interpolation** : Sélectionnez une méthode d'interpolation : Interpolation Triangulaire (TIN) ... ou Pondération par Distance Inverse (IDW) ... .
– **Nombre de colonnes/lignes** : Définit le nombre de lignes et de colonnes du raster de sortie.
– **Fichier de sortie** : Définit un nom pour le fichier raster de sortie.

FIG. 52: Interpolation Plugin



1. Lancez QGIS et chargez le tableau CSV `elevp.csv` contenant les points d'élévations dans le canevas de QGIS en utilisant l'extension décrite dans la section 12.3.

2. Chargez l'extension d'Interpolation (voir la section 11.1.1) et cliquez sur le bouton Interpolation qui apparaît dans le menu de la barre d'outils. Un nouveau dialogue se présente comme l'illustre la figure 52.

3. Sélectionnez elevp ... comme entrée de vecteur et la colonne `ELEV` pour l'interpolation.

4. Sélectionnez Interpolation Triangulaire ... comme méthode d'interpolation, définissez

3663 colonnes et 1964 lignes (ce qui équivaut à une résolution par pixel de 1000 mètres) et `elevation_tin` comme nom pour le fichier en sortie.

5. Cliquez sur [Ok].
6. Double-cliquez sur `elevation_tin` dans la légende de la carte pour ouvrir la fenêtre de propriétés de la couche raster et sélectionnez [Pseudocouleur ... ▼] comme couleurs pour la carte dans l'onglet [Symbologie]. Vous pouvez aussi définir une nouvelle table de couleurs comme expliquée dans la section 6.3.

Dans la figure 53 vous pouvez voir le résultat d'une interpolation IDW avec une résolution de 366 colonnes x 196 lignes (soit 10 km), pour les données du fichier `elevp.csv`. Le calcul peut prendre plusieurs minutes même si les données ne couvrent que la partie nord de l'Alaska.

FIG. 53: Interpolation d'une carte d'élévation en utilisant la méthode IDW

12.11 Extension d'exportation Mapserver

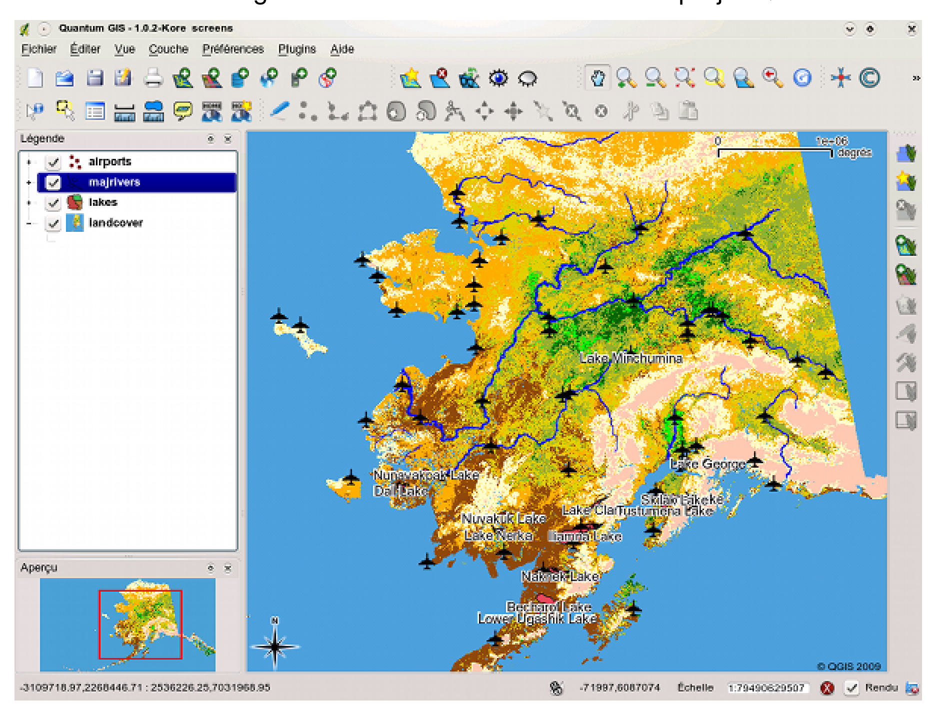
Vous pouvez utiliser QGIs pour "composer" votre carte en ajoutant et en arrangeant des couches, en modifiant leur représentation graphique, puis vous pouvez exporter le résultat sous la forme d'un fichier .map à destination de MapServer. Pour utiliser cette extension, vous devez avoir Python >= 2.4 installé sur votre système ainsi qu'une version de QGIS compilée avec le support adéquat. Toutes les versions officielles disponibles offrent ce support de Python.

L'extension d'exportation Mapserver de QGIS 1.0.0 est une extension python, qui est automatiquement chargée dans le `Gestionnaire d'extension` comme l'une des extensions principales (voir Section 12).

12.11.1 Création du fichier de projet

L'extension fonctionne sur un fichier de projet QGIS précédemment enregistré, et non **pas** sur le contenu actuel de la carte et de la légende. C'est souvent une source de confusion pour beaucoup. Comme décrit ci-dessous, vous avez besoin de réarranger les couches vecteurs et rasters que vous voulez utiliser dans MapServer et enregistrer l'état qui paraît satisfaisant dans un fichier de projet QGIS.

FIG. 54: Arrangement des couches d'un fichier de projet QGIS



Dans cet exemple sont présentées les quatres étapes menant à la création du fichier .map pour Map-

Server. Les fichiers vecteurs et rasters proviennent de l'échantillon de jeu de données de Quantum GIS 3.2.

1. Ajoutez la couche raster `landcover.tif` en cliquant sur l'icône  Ajouter une Couche Raster.
2. Ajoutez la couche vecteur des shapefiles `lakes.shp`, `majrivers.shp` et `airports.shp` depuis le jeu de données en cliquant sur l'icône Ajouter une couche Vecteur.
3. Changer les couleurs et les symboles des données (voir Figure 54)
4. Enregistrez dans un nouveau fichier de projet nommé `mapserverproject.qgs` en utilisant File > Enregistrer le projet.

12.11.2 Création du fichier .map

L'outil `msexport` d'exportation se situe dans votre répertoire d'installation de QGIS et peut être utilisé indépendamment de QGIS. Pour l'utiliser à partir du logiciel vous devez charger l'extension en utilisant le Gestionaire d'extension. Cliquez sur Plugins > Gestionnaire d'extension pour l'ouvrir, sélectionnez l'extension d'exportation vers MapServer et faites OK . Maintenant lancez le dialogue



Exporter vers MapServer (voir Figure 55) en cliquant sur l'icône dans la barre de menu.

Fichier .map
: Saisissez le chemin complet du fichier .map que vous voulez exporter. Vous pouvez utiliser le bouton sur la droite pour parcourir votre système.

Fichier projet Qgis
: Saisissez le chemin complet du fichier projet (.qgs) que vous voulez exporter. Vous pouvez utiliser le bouton sur la droite pour parcourir votre système.

Nom de la carte
: Un nom pour la carte. Ce nom préfixera toutes les images créées par le serveur.

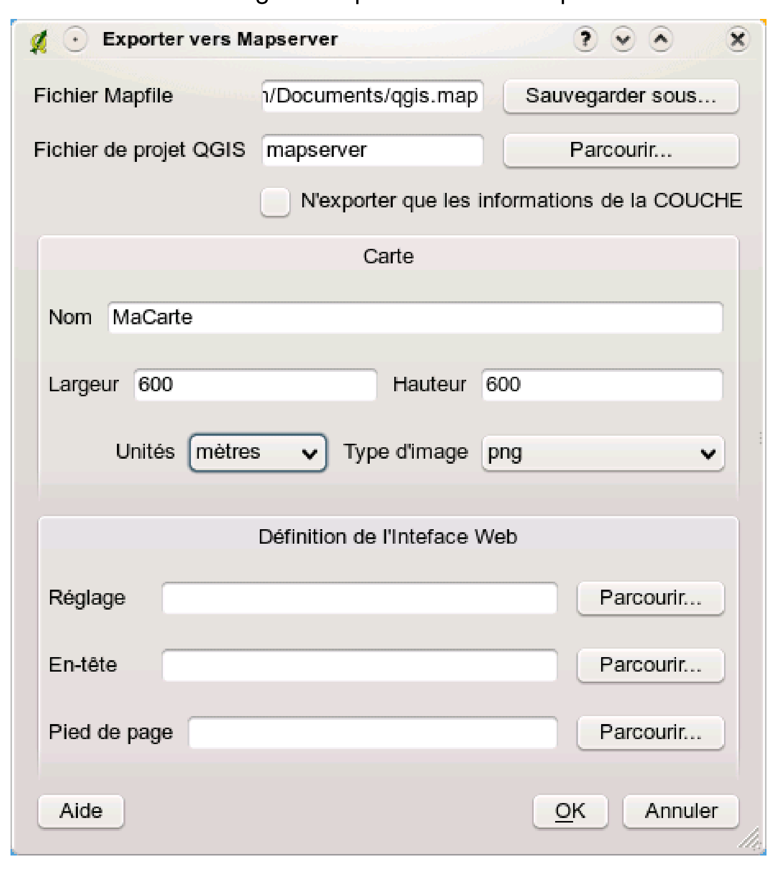
Largeur de la carte
: Largeur en pixels de l'image générée.

Hauteur de la carte
: hauteur en pixels de l'image générée.

Unitées de la carte
: Unités de mesure utilisées

Type d'image
: Format de l'image générée par MapServer

FIG. 55: Dialogue d'exportation vers MapServer



Web Template
: Chemin complet vers le fichier MapServer template à utiliser

Web en-tête
: Chemin complet vers le fichier d'en-tête de MapServer à utiliserMapServer

Web bas de page
: Chemin complet vers le fichier de bas de page MapServer à utiliser

Seulement le `fichier .map` et le `fichier de projet QGIS` sont requis pour créer un fichier .map, vous risquez cependant d'obtenir un fichier inutilisable selon ce que vous désirez en faire. Bien que QGIS soit en mesure de créer les fichiers .map, votre projet nécessite peut-être des adaptions pour obtenir le résultat escompté. Créons un fichier .map en utilisant le projet `mapserverproject.qgs` que nous avons créé (voir Figure 55) :

1. Ouvrez l'extension en cliquant sur l'icône Exporter vers.
2. Entrez le nom `qgisproject.map` pour votre nouveau fichier .map.

3. Sélectionnez le projet QGIS `mapserverproject.qgs` que vous venez d'enregistrer.
4. Entrez un nom `MaCarte` pour la carte.
5. Entrez `600` pour la largeur et `400` pour la hauteur.
6. Nos couches sont en mètres, l'unité de mesure sera donc le mètre.
7. Choisissez "png" pour le type d'image.
8. Cliquez sur OK pour générer le ouveau fichier `qgisproject.map`. QGIS affiche le résultat de vos efforts.

Vous pouvez visualiser le fichier .map dans un éditeur de texte. Si vous le faites, vous remarquerez que l'outil d'exportation ajoute les métadonnées nécessaires à l'utilisation du protocole WMS.

12.11.3 Essai du fichier .map

Nous pouvons maintenant tester notre travail en utilisant l'outil `shp2img` pour créer une image tirée du fichier .map. Cet outil est présent dans MapServer et FWTools. Pour créer une image :

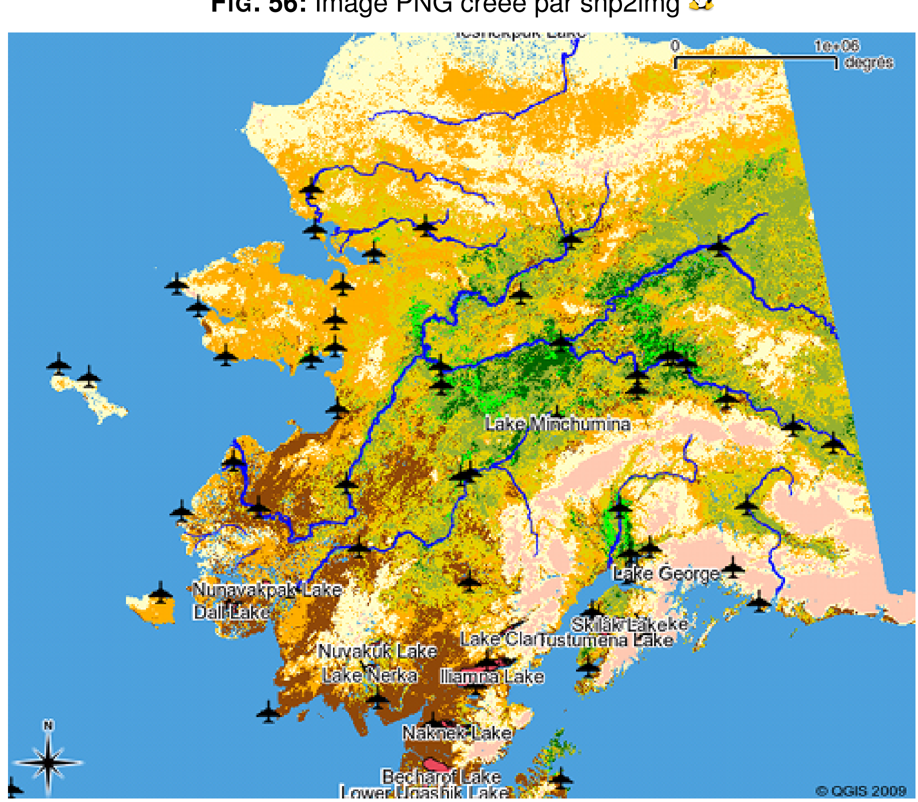
– Ouvrez une console de commande
– Si vous n'avez pas sauvegardé votre fichier dans votre répertoire personnel, déplacer vers le bon dossier
– Lancez `shp2img -m qgisproject.map -o mapserver_test.png` et affichez l'image

Une image PNG est créée avec toutes les couches du projet QGIS. En complément, l'étendue du PNG sera la même que celle du projet. Comme vous le voyez sur la figure 56, toutes les informations, à l'exception des symboles des aéroports, sont inclues.

Si vous comptez utiliser ce fichier .map pour fournir un service WMS, vous n'aurez probablement rien d'autres à modifier. Par contre si vous comptez l'utiliser comme modèle ou dans une interface personnalisée vous aurez un peu plus de travaux manuels à effectuer. Vous pouvez jeter un oeil sur cette vidéo de Christopher Schmidt (version 0.8 de QGIS mais cela reste valable). [10]

[10] http://openlayers.org/presentations/mappingyourdata/

FIG. 56: Image PNG créée par shp2img

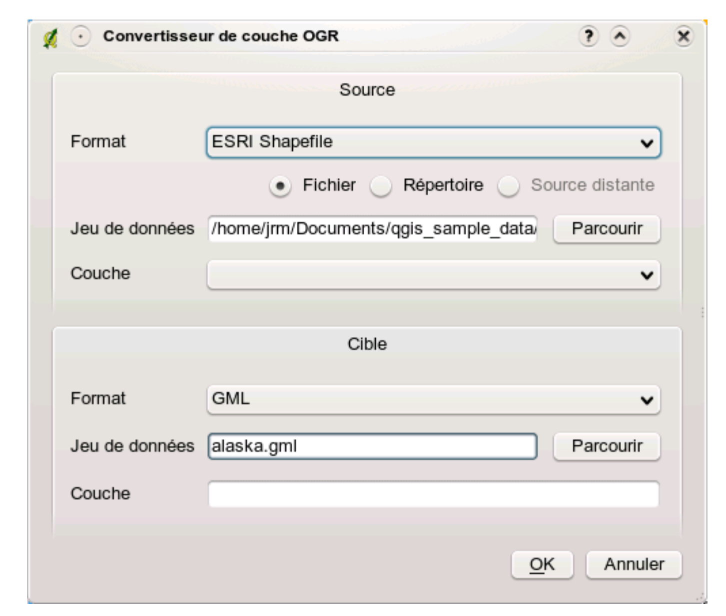


12.12 Extension OGR pour la conversion

L'extension de conversion de couche OGR vous permet de convertir des données vecteurs d'un format supporté par la librairie OGR vers un autre format supporté par la librairie OGR. Il est très simple à manipuler et vous donne accès à des fonctionnalités comme montré sur la Figure 57. Les formats supportés peuvent varier en fonction de la version GDAL/OGR installée.

– **Source Format/Datset/Layer** : Choisissez le format OGR et donnez le chemin vers le fichier à convertir
– **Target Format/Datset/Layer** : Choisissez le format OGR de sortie ainsi que l'emplacement du fichier de sortie

FIG. 57: Convertisseur de couche OGR



1. Démarrez QGIS, chargez le convertisseur OGR dans le gestionnaire de plugin (see Section 11.1.1) et cliquez sur le bouton OGR Layer Converter qui se trouve dans la barre de menu QGIS. La boite de dialogue convertisseur OGR apparait comme sur la figure 57.
2. Sélectionnez le format OGR ESRI Shapefile ... et le chemin vers le fichier d'entrée `alaska.shp` dans la partie Source de la boite de dialogue.
3. Sélectionnez le format OGR GML ... et donnez le chemin ainsi que le nom du fichier de sortie `alaska.gml` dans la partie Cible de la boite de dialogue.
4. Cliquez sur Ok .

13 Utiliser des extensions externes en Python pour QGIS

Les extensions QGIS externes sont écrites en python. Elles sont stockées dans un dépôt officiel, modéré et maintenu par l'auteur initial. Le tableau 7 montre une liste d'extensions disponibles actuellement avec une courte description.[11] [12]

Quand ce manuel a été publié, le dépôt modéré d'extensions externes de QGIS n'était pas encore bien établi. Une documentation détaillée sur l'utilisation, l'auteur et d'autres informations importantes sont fournis avec l'extension externe et ne font pas partie de ce manuel.

Vous trouverez une liste à jour des plugins externes modérés dans le dépôt officiel de QGIS du menu Récupérer les extensionss pythons ... et sur `http://qgis.osgeo.org/download/plugins.html`.

TAB. 7: Extensions QGIS externe actuellement modéré

Icône	Extension externe	Description
	Zoom vers un point	Zooms vers des coordonnées définies dans la boîte de dialogue. Vous pouvez définir également le niveau de zoom pour contrôler l'étendue de la vue.

Une description de l'installation détaillée pour les extensions python externe peut être trouvée dans la section 11.1.2.

Dépôt d'extensions Python réalisées par les utilisateurs et les dépôts des auteurs

En plus des extensions externes modérées, il existe d'autres dépôts non officiels d' extensions Python. Il contient des extensions qui ne sont pas assez matures pour être inclus au dépôt officiel. Cependant, certains peuvent être très utiles. De plus, certains de nos contributeurs maintiennent leurs propres dépôts.

Pour ajouter des dépôts non officiels et des dépôts des auteurs, ouvrez l'installateur d'extensions (Plugins > Récupérer les extensions python ...), allez sur l'onglet Dépôt et cliquez sur le bouton Ajouter un dépôt supplémentaire . Si vous ne voulez pas un ou plusieurs dépôts ajoutés, désactivez-le avec le bouton Éditer ... ou supprimez-le complètement avec le bouton Supprimer .

Astuce 43 AJOUTER D'AUTRES EXTENSIONS EXTERNES
en plus du dépôt officiel d'extensions de QGIS vous pouvez ajouter d'autres dépôts externes. Pour cela, sélectionnez l'onglet dépôt dans l'installateur d'extensions Python.

[11] Les mises à jour des extensions principales peuvent également être disponibles via ce dépôt
[12] L'installateur d'extensions python est également une extension python externe. Mais il fait partie des sources de QGIS et est automatiquement chargé et sélectionné dans le gestionnaire d'extension de QGIS (voir la section 11.1.2).

14 Écrire des extensions pour QGIS en C++

Dans cette section nous fournissons un tutoriel pour débutant pour l'écriture d' extension simple en C++ pour QGIS. Il est basé sur le workshop réalisé par Dr. Marco Hugentobler.

Les extensions C++ de QGIS sont des bibliothèques dynamiquement liées (.so ou .dll). Elles sont liées à QGIS pendant son fonctionnement à la demande dans le gestionnaire d'extension et étendent les fonctionnalités de QGIS. Ils ont accès à l'interface de QGIS et peuvent être divisés en extensions principales et externes.

Techniquement le gestionnaire d'extension de QGIS cherche tous les fichiers .so dans le répertoire lib/qgis et les charge lors du démarrage. Lors de la fermeture, ils sont déchargés, sauf ceux avec une case cochée. Pour les plugins nouvellement chargés, la méthode *classFactory* du plugin est appelée pour afficher l'interface de l'extension dans le menu extension et la barre d'outils. La fonction *unload()* de l' est utilisée pour enlever les éléments de l'interface allouée et la classe d'extension est enlevée en utilisant le destructeur de classe. Pour lister les extensions, chaque extension doit avoir quelques fonctions 'C' externes pour la description et bien sur la méthode *classFactory*.

14.1 Pourquoi C++ et quelle licence est utilisée

QGIS lui-même est écrit en C++, il est donc logique d'écrire des extensions en C++. C'est un langage orienté-objet (OOP) qui est considéré par beaucoup de développeurs comme langage à préférer pour la création d'applications de taille importantes.

Les extensions C++ de QGIS utilisent des fonctionnalités des bibliothèques libqgis*.so. Comme elles sont sous licence GNU GPL, les extensions C++ de QGIS doivent être aussi sous cette licence. Cela signifie que vous pouvez utiliser vos extensions comme vous le souhaitez et que vous n'êtes pas obligé de les publier. Cependant si vous les publiez, vous devez les publier sous les conditions de la licence GPL.

14.2 Programmer une extension en C++ pour QGIS en quatre étapes

L'extension d'exemple est une extension de conversion de point et reste simple intentionnellement. L'extension recherche la couche vecteur active dans QGIS, convertit tous les sommets des géométries de la couche en objets ponctuels en gardant les attributs et enfin écrit les objets ponctuels dans un fichier csv. La nouvelle couche peut alors être chargée dans QGIS en utilisant l'extension de délimitation de texte (voir section 12.3).

Étape 1 : Permettre au gestionnaire d'extensions de reconnaitre l'extension

Pour commencer nous créons les fichiers `QgsPointConverter.h` et `QgsPointConverter.cpp`. Puis nous ajoutons les méthodes héritées de QgisPlugin (mais les laissons vides pour l'instant), créons les méthodes 'C' externes nécessaires et un fichier .pro, qui est un mécanisme de Qt pour créer facilement un Makefiles. Puis nous compilons les sources, déplaçons la bibliothèque compilée dans le répertoire des extensions et le chargeons dans le gestionnaire d'extensions de QGIS.

a) Créez un nouveau fichier pointconverter.pro et ajoutez :

```
#base directory of the qgis installation
QGIS_DIR = /home/marco/src/qgis

TEMPLATE = lib
CONFIG = qt
QT += xml qt3support
unix:LIBS += -L/$$QGIS_DIR/lib -lqgis_core -lqgis_gui
INCLUDEPATH += $$QGIS_DIR/src/ui $$QGIS_DIR/src/plugins  $$QGIS_DIR/src/gui \
        $$QGIS_DIR/src/raster $$QGIS_DIR/src/core $$QGIS_DIR
SOURCES = qgspointconverterplugin.cpp
HEADERS = qgspointconverterplugin.h
DEST = pointconverterplugin.so
DEFINES += GUI_EXPORT= CORE_EXPORT=
```

b) Créez un nouveau fichier qgspointconverterplugin.h et ajoutez :

```
#ifndef QGSPOINTCONVERTERPLUGIN_H
#define QGSPOINTCONVERTERPLUGIN_H

#include "qgisplugin.h"

/**A plugin that converts vector layers to delimited text point files.
 The vertices of polygon/line type layers are converted to point features*/
class QgsPointConverterPlugin: public QgisPlugin
{
  public:
  QgsPointConverterPlugin(QgisInterface* iface);
  ~QgsPointConverterPlugin();
  void initGui();
  void unload();

  private:
```

```
  QgisInterface* mIface;
};
#endif
```

b) Créez un nouveau fichier qgspointconverterplugin.cpp et ajoutez :

```
#include "qgspointconverterplugin.h"

#ifdef WIN32
#define QGISEXTERN extern "C" __declspec( dllexport )
#else
#define QGISEXTERN extern "C"
#endif

QgsPointConverterPlugin::QgsPointConverterPlugin(QgisInterface* iface): mIface(iface)
{
}

QgsPointConverterPlugin::~QgsPointConverterPlugin()
{
}

void QgsPointConverterPlugin::initGui()
{
}

void QgsPointConverterPlugin::unload()
{
}

QGISEXTERN QgisPlugin* classFactory(QgisInterface* iface)
{
  return new QgsPointConverterPlugin(iface);
}

QGISEXTERN QString name()
{
  return "point converter plugin";
}

QGISEXTERN QString description()
```

```
{
  return "A plugin that converts vector layers to delimited text point files";
}

QGISEXTERN QString version()
{
  return "0.00001";
}

// Return the type (either UI or MapLayer plugin)
QGISEXTERN int type()
{
  return QgisPlugin::UI;
}

// Delete ourself
QGISEXTERN void unload(QgisPlugin* theQgsPointConverterPluginPointer)
{
  delete theQgsPointConverterPluginPointer;
}
```

Étape 2 : Créer un icône et un menu pour le plugin

Cette étape inclut l'ajout d'un pointeur à l'objet QgisInterface dans la classe d'extension. Puis nous créons une fonction QAction et un callback (slot), l'ajoutons à l'interface de QGIS en utilisant QgisIface : :addToolBarIcon() et QgisIface : :addPluginToMenu() et enfin nous enlevons QAction dans la méthode *unload()*.

d) Ouvrez de nouveau le fichier qgspointconverterplugin.h et ajoutez au contenu existant :

```
#ifndef QGSPOINTCONVERTERPLUGIN_H
#define QGSPOINTCONVERTERPLUGIN_H

#include "qgisplugin.h"
#include <QObject>

class QAction;

/**A plugin that converts vector layers to delimited text point files.
 The vertices of polygon/line type layers are converted to point features*/
class QgsPointConverterPlugin: public QObject, public QgisPlugin
{
```

```
  Q_OBJECT

 public:
  QgsPointConverterPlugin(QgisInterface* iface);
  ~QgsPointConverterPlugin();
  void initGui();
  void unload();

 private:
  QgisInterface* mIface;
  QAction* mAction;

   private slots:
   void convertToPoint();
};

#endif
```

e) Ouvrez de nouveau le fichier qgspointconverterplugin.cpp et ajoutez au contenu existant :

```
#include "qgspointconverterplugin.h"
#include "qgisinterface.h"
#include <QAction>

#ifdef WIN32
#define QGISEXTERN extern "C" __declspec( dllexport )
#else
#define QGISEXTERN extern "C"
#endif

QgsPointConverterPlugin::QgsPointConverterPlugin(QgisInterface* iface): \
    mIface(iface), mAction(0)
{

}

QgsPointConverterPlugin::~QgsPointConverterPlugin()
{

}
```

```
void QgsPointConverterPlugin::initGui()
{
  mAction = new QAction(tr("&Convert to point"), this);
  connect(mAction, SIGNAL(activated()), this, SLOT(convertToPoint()));
  mIface->addToolBarIcon(mAction);
  mIface->addPluginToMenu(tr("&Convert to point"), mAction);
}

void QgsPointConverterPlugin::unload()
{
  mIface->removeToolBarIcon(mAction);
  mIface->removePluginMenu(tr("&Convert to point"), mAction);
  delete mAction;
}

void QgsPointConverterPlugin::convertToPoint()
{
  qWarning("in method convertToPoint");
}

QGISEXTERN QgisPlugin* classFactory(QgisInterface* iface)
{
  return new QgsPointConverterPlugin(iface);
}

QGISEXTERN QString name()
{
  return "point converter plugin";
}

QGISEXTERN QString description()
{
  return "A plugin that converts vector layers to delimited text point files";
}

QGISEXTERN QString version()
{
  return "0.00001";
}

// Return the type (either UI or MapLayer plugin)
QGISEXTERN int type()
```

```
{
  return QgisPlugin::UI;
}

// Delete ourself
QGISEXTERN void unload(QgisPlugin* theQgsPointConverterPluginPointer)
{
  delete theQgsPointConverterPluginPointer;
}
```

Étape 3 : Lire des géométries ponctuelles à partir de la couche active et écrire dans un fichier texte

Pour lire la géométrie ponctuelle à partir de la couche active nous devons demander la couche en court et la localisation du fichier texte. Puis nous itérons sur toutes les géométries de la couche actuelle, convertissons les géométries (sommets) en points, ouvrons un nouveau fichier et utilisons QTextStream pour écrire les coordonnées x et y à l'intérieur.

f) Ouvrez de nouveau qgspointconverterplugin.h et ajoutez ceci au contenu existant

```
class QgsGeometry;
class QTextStream;

private:

void convertPoint(QgsGeometry* geom, const QString& attributeString, \
    QTextStream& stream) const;
void convertMultiPoint(QgsGeometry* geom, const QString& attributeString, \
    QTextStream& stream) const;
void convertLineString(QgsGeometry* geom, const QString& attributeString, \
    QTextStream& stream) const;
void convertMultiLineString(QgsGeometry* geom, const QString& attributeString, \
    QTextStream& stream) const;
void convertPolygon(QgsGeometry* geom, const QString& attributeString, \
    QTextStream& stream) const;
void convertMultiPolygon(QgsGeometry* geom, const QString& attributeString, \
    QTextStream& stream) const;
```

g) Ouvrez qgspointconverterplugin.cpp et rajoutez ceci au contenu existant :

```
#include "qgsgeometry.h"
```

```
#include "qgsvectordataprovider.h"
#include "qgsvectorlayer.h"
#include <QFileDialog>
#include <QMessageBox>
#include <QTextStream>

void QgsPointConverterPlugin::convertToPoint()
{
  qWarning("in method convertToPoint");
  QgsMapLayer* theMapLayer = mIface->activeLayer();
  if(!theMapLayer)
    {
      QMessageBox::information(0, tr("no active layer"), \
      tr("this plugin needs an active point vector layer to make conversions \
          to points"), QMessageBox::Ok);
      return;
    }
  QgsVectorLayer* theVectorLayer = dynamic_cast<QgsVectorLayer*>(theMapLayer);
  if(!theVectorLayer)
    {
      QMessageBox::information(0, tr("no vector layer"), \
      tr("this plugin needs an active point vector layer to make conversions \
          to points"), QMessageBox::Ok);
      return;
    }

  QString fileName = QFileDialog::getSaveFileName();
  if(!fileName.isNull())
    {
      qWarning("The selected filename is: " + fileName);
      QFile f(fileName);
      if(!f.open(QIODevice::WriteOnly))
      {
 QMessageBox::information(0, "error", "Could not open file", QMessageBox::Ok);
 return;
      }
      QTextStream theTextStream(&f);
      theTextStream.setRealNumberNotation(QTextStream::FixedNotation);

      QgsFeature currentFeature;
      QgsGeometry* currentGeometry = 0;
```

```
    QgsVectorDataProvider* provider = theVectorLayer->dataProvider();
    if(!provider)
    {
        return;
    }

    theVectorLayer->select(provider->attributeIndexes(), \
    theVectorLayer->extent(), true, false);

    //write header
    theTextStream << "x,y";
    theTextStream << endl;

    while(theVectorLayer->nextFeature(currentFeature))
    {
QString featureAttributesString;

      currentGeometry = currentFeature.geometry();
      if(!currentGeometry)
      {
          continue;
      }

      switch(currentGeometry->wkbType())
      {
          case QGis::WKBPoint:
          case QGis::WKBPoint25D:
              convertPoint(currentGeometry, featureAttributesString, \
theTextStream);
              break;

          case QGis::WKBMultiPoint:
          case QGis::WKBMultiPoint25D:
              convertMultiPoint(currentGeometry, featureAttributesString, \
theTextStream);
              break;

          case QGis::WKBLineString:
          case QGis::WKBLineString25D:
              convertLineString(currentGeometry, featureAttributesString, \
theTextStream);
              break;
```

```
            case QGis::WKBMultiLineString:
            case QGis::WKBMultiLineString25D:
                convertMultiLineString(currentGeometry, featureAttributesString \
  theTextStream);
                break;

            case QGis::WKBPolygon:
            case QGis::WKBPolygon25D:
                convertPolygon(currentGeometry, featureAttributesString, \
  theTextStream);
                break;

            case QGis::WKBMultiPolygon:
            case QGis::WKBMultiPolygon25D:
                convertMultiPolygon(currentGeometry, featureAttributesString, \
  theTextStream);
                break;
        }
      }
    }
}

//geometry converter functions
void QgsPointConverterPlugin::convertPoint(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsPoint p = geom->asPoint();
    stream << p.x() << "," << p.y();
    stream << endl;
}

void QgsPointConverterPlugin::convertMultiPoint(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsMultiPoint mp = geom->asMultiPoint();
    QgsMultiPoint::const_iterator it = mp.constBegin();
    for(; it != mp.constEnd(); ++it)
    {
        stream << (*it).x() << "," << (*it).y();
        stream << endl;
    }
```

```
}

void QgsPointConverterPlugin::convertLineString(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsPolyline line = geom->asPolyline();
    QgsPolyline::const_iterator it = line.constBegin();
    for(; it != line.constEnd(); ++it)
    {
        stream << (*it).x() << "," << (*it).y();
        stream << endl;
    }
}

void QgsPointConverterPlugin::convertMultiLineString(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsMultiPolyline ml = geom->asMultiPolyline();
    QgsMultiPolyline::const_iterator lineIt = ml.constBegin();
    for(; lineIt != ml.constEnd(); ++lineIt)
    {
        QgsPolyline currentPolyline = *lineIt;
        QgsPolyline::const_iterator vertexIt = currentPolyline.constBegin();
        for(; vertexIt != currentPolyline.constEnd(); ++vertexIt)
        {
            stream << (*vertexIt).x() << "," << (*vertexIt).y();
            stream << endl;
        }
    }
}

void QgsPointConverterPlugin::convertPolygon(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsPolygon polygon = geom->asPolygon();
    QgsPolygon::const_iterator it = polygon.constBegin();
    for(; it != polygon.constEnd(); ++it)
    {
        QgsPolyline currentRing = *it;
        QgsPolyline::const_iterator vertexIt = currentRing.constBegin();
        for(; vertexIt != currentRing.constEnd(); ++vertexIt)
        {
```

```
            stream << (*vertexIt).x() << "," << (*vertexIt).y();
            stream << endl;
        }
    }
}

void QgsPointConverterPlugin::convertMultiPolygon(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsMultiPolygon mp = geom->asMultiPolygon();
    QgsMultiPolygon::const_iterator polyIt = mp.constBegin();
    for(; polyIt != mp.constEnd(); ++polyIt)
    {
        QgsPolygon currentPolygon = *polyIt;
        QgsPolygon::const_iterator ringIt = currentPolygon.constBegin();
        for(; ringIt != currentPolygon.constEnd(); ++ringIt)
        {
            QgsPolyline currentPolyline = *ringIt;
            QgsPolyline::const_iterator vertexIt = currentPolyline.constBegin();
            for(; vertexIt != currentPolyline.constEnd(); ++vertexIt)
            {
                stream << (*vertexIt).x() << "," << (*vertexIt).y();
                stream << endl;
            }
        }
    }
}
```

Étape 4 : Copier les attributs des géométries dans le fichier texte

Finalement, nous récuperons les attributs de la couche active en utilisant QgsVectorDataProvider : :fieldNameMap(). Pour chaque géométrie nous récupérons les valeurs des champs en utilisant QgsFeature : :attributeMap() et ajoutons le contenu séparé d'une virgule après les coordonnées x et y pour chaque nouvel objet ponctuel. Pour cette étape il n'y a pas besoin de changement supplémentaire dans le fichier `qgspointconverterplugin.h`.

h) Ouvrez de nouveau le fichier qgspointconverterplugin.cpp et rajoutez ceci au contenu :

```
#include "qgspointconverterplugin.h"
#include "qgisinterface.h"
#include "qgsgeometry.h"
#include "qgsvectordataprovider.h"
```

```
#include "qgsvectorlayer.h"
#include <QAction>
#include <QFileDialog>
#include <QMessageBox>
#include <QTextStream>

#ifdef WIN32
#define QGISEXTERN extern "C" __declspec( dllexport )
#else
#define QGISEXTERN extern "C"
#endif

QgsPointConverterPlugin::QgsPointConverterPlugin(QgisInterface* iface): \
mIface(iface), mAction(0)
{

}

QgsPointConverterPlugin::~QgsPointConverterPlugin()
{

}

void QgsPointConverterPlugin::initGui()
{
  mAction = new QAction(tr("&Convert to point"), this);
  connect(mAction, SIGNAL(activated()), this, SLOT(convertToPoint()));
  mIface->addToolBarIcon(mAction);
  mIface->addPluginToMenu(tr("&Convert to point"), mAction);
}

void QgsPointConverterPlugin::unload()
{
  mIface->removeToolBarIcon(mAction);
  mIface->removePluginMenu(tr("&Convert to point"), mAction);
  delete mAction;
}

void QgsPointConverterPlugin::convertToPoint()
{
  qWarning("in method convertToPoint");
  QgsMapLayer* theMapLayer = mIface->activeLayer();
```

```
  if(!theMapLayer)
    {
      QMessageBox::information(0, tr("no active layer"), \
      tr("this plugin needs an active point vector layer to make conversions \
          to points"), QMessageBox::Ok);
      return;
    }
  QgsVectorLayer* theVectorLayer = dynamic_cast<QgsVectorLayer*>(theMapLayer);
  if(!theVectorLayer)
    {
      QMessageBox::information(0, tr("no vector layer"), \
      tr("this plugin needs an active point vector layer to make conversions \
          to points"), QMessageBox::Ok);
      return;
    }

  QString fileName = QFileDialog::getSaveFileName();
  if(!fileName.isNull())
    {
      qWarning("The selected filename is: " + fileName);
      QFile f(fileName);
      if(!f.open(QIODevice::WriteOnly))
      {
QMessageBox::information(0, "error", "Could not open file", QMessageBox::Ok);
return;
      }
      QTextStream theTextStream(&f);
      theTextStream.setRealNumberNotation(QTextStream::FixedNotation);

      QgsFeature currentFeature;
      QgsGeometry* currentGeometry = 0;

      QgsVectorDataProvider* provider = theVectorLayer->dataProvider();
      if(!provider)
      {
          return;
      }

      theVectorLayer->select(provider->attributeIndexes(), \
      theVectorLayer->extent(), true, false);

      //write header
```

```
theTextStream << "x,y";
QMap<QString, int> fieldMap = provider->fieldNameMap();
//We need the attributes sorted by index.
//Therefore we insert them in a second map where key / values are exchanged
QMap<int, QString> sortedFieldMap;
QMap<QString, int>::const_iterator fieldIt = fieldMap.constBegin();
for(; fieldIt != fieldMap.constEnd(); ++fieldIt)
{
  sortedFieldMap.insert(fieldIt.value(), fieldIt.key());
}

QMap<int, QString>::const_iterator sortedFieldIt = sortedFieldMap.constBegin();
for(; sortedFieldIt != sortedFieldMap.constEnd(); ++sortedFieldIt)
{
    theTextStream << "," << sortedFieldIt.value();
}

theTextStream << endl;

while(theVectorLayer->nextFeature(currentFeature))
{
  QString featureAttributesString;
   const QgsAttributeMap& map = currentFeature.attributeMap();
   QgsAttributeMap::const_iterator attributeIt = map.constBegin();
   for(; attributeIt != map.constEnd(); ++attributeIt)
   {
      featureAttributesString.append(",");
      featureAttributesString.append(attributeIt.value().toString());
   }


  currentGeometry = currentFeature.geometry();
  if(!currentGeometry)
  {
      continue;
  }

  switch(currentGeometry->wkbType())
  {
      case QGis::WKBPoint:
      case QGis::WKBPoint25D:
          convertPoint(currentGeometry, featureAttributesString, \
```

```
theTextStream);
                break;

            case QGis::WKBMultiPoint:
            case QGis::WKBMultiPoint25D:
                convertMultiPoint(currentGeometry, featureAttributesString, \
theTextStream);
                break;

            case QGis::WKBLineString:
            case QGis::WKBLineString25D:
                convertLineString(currentGeometry, featureAttributesString, \
theTextStream);
                break;

            case QGis::WKBMultiLineString:
            case QGis::WKBMultiLineString25D:
                convertMultiLineString(currentGeometry, featureAttributesString \
theTextStream);
                break;

            case QGis::WKBPolygon:
            case QGis::WKBPolygon25D:
                convertPolygon(currentGeometry, featureAttributesString, \
theTextStream);
                break;

            case QGis::WKBMultiPolygon:
            case QGis::WKBMultiPolygon25D:
                convertMultiPolygon(currentGeometry, featureAttributesString, \
theTextStream);
                break;
        }
      }
    }
}

//geometry converter functions
void QgsPointConverterPlugin::convertPoint(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsPoint p = geom->asPoint();
```

```
    stream << p.x() << "," << p.y();
    stream << attributeString;
    stream << endl;
}

void QgsPointConverterPlugin::convertMultiPoint(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsMultiPoint mp = geom->asMultiPoint();
    QgsMultiPoint::const_iterator it = mp.constBegin();
    for(; it != mp.constEnd(); ++it)
    {
        stream << (*it).x() << "," << (*it).y();
        stream << attributeString;
        stream << endl;
    }
}

void QgsPointConverterPlugin::convertLineString(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsPolyline line = geom->asPolyline();
    QgsPolyline::const_iterator it = line.constBegin();
    for(; it != line.constEnd(); ++it)
    {
        stream << (*it).x() << "," << (*it).y();
        stream << attributeString;
        stream << endl;
    }
}

void QgsPointConverterPlugin::convertMultiLineString(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsMultiPolyline ml = geom->asMultiPolyline();
    QgsMultiPolyline::const_iterator lineIt = ml.constBegin();
    for(; lineIt != ml.constEnd(); ++lineIt)
    {
        QgsPolyline currentPolyline = *lineIt;
        QgsPolyline::const_iterator vertexIt = currentPolyline.constBegin();
        for(; vertexIt != currentPolyline.constEnd(); ++vertexIt)
        {
```

```
            stream << (*vertexIt).x() << "," << (*vertexIt).y();
            stream << attributeString;
            stream << endl;
        }
    }
}

void QgsPointConverterPlugin::convertPolygon(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsPolygon polygon = geom->asPolygon();
    QgsPolygon::const_iterator it = polygon.constBegin();
    for(; it != polygon.constEnd(); ++it)
    {
        QgsPolyline currentRing = *it;
        QgsPolyline::const_iterator vertexIt = currentRing.constBegin();
        for(; vertexIt != currentRing.constEnd(); ++vertexIt)
        {
            stream << (*vertexIt).x() << "," << (*vertexIt).y();
            stream << attributeString;
            stream << endl;
        }
    }
}

void QgsPointConverterPlugin::convertMultiPolygon(QgsGeometry* geom, const QString& \
attributeString, QTextStream& stream) const
{
    QgsMultiPolygon mp = geom->asMultiPolygon();
    QgsMultiPolygon::const_iterator polyIt = mp.constBegin();
    for(; polyIt != mp.constEnd(); ++polyIt)
    {
        QgsPolygon currentPolygon = *polyIt;
        QgsPolygon::const_iterator ringIt = currentPolygon.constBegin();
        for(; ringIt != currentPolygon.constEnd(); ++ringIt)
        {
            QgsPolyline currentPolyline = *ringIt;
            QgsPolyline::const_iterator vertexIt = currentPolyline.constBegin();
            for(; vertexIt != currentPolyline.constEnd(); ++vertexIt)
            {
                stream << (*vertexIt).x() << "," << (*vertexIt).y();
                stream << attributeString;
```

```
                stream << endl;
            }
        }
    }
}

QGISEXTERN QgisPlugin* classFactory(QgisInterface* iface)
{
  return new QgsPointConverterPlugin(iface);
}

QGISEXTERN QString name()
{
  return "point converter plugin";
}

QGISEXTERN QString description()
{
  return "A plugin that converts vector layers to delimited text point files";
}

QGISEXTERN QString version()
{
  return "0.00001";
}

// Return the type (either UI or MapLayer plugin)
QGISEXTERN int type()
{
  return QgisPlugin::UI;
}

// Delete ourself
QGISEXTERN void unload(QgisPlugin* theQgsPointConverterPluginPointer)
{
  delete theQgsPointConverterPluginPointer;
}
```

14.3 Plus d'informations

Comme vous pouvez le voir, vous avez besoin d'informations à partir de différentes sources pour écrire des extensions C++ pour QGIS. Les développeurs d'extensions doivent connaître le C++, l'interface des extensions de QGIS ainsi que les classes et outils Qt4. Au début il est plus enrichissant d'apprendre à partir des exemples et de copier les mécanismes d'extensions existants.

Il y a un certain nombre de documentations qui peut être utile pour les programmeurs C++ pour QGIS :

- Debuguer des extensions de QGIS : `http://wiki.qgis.org/qgiswiki/DebuggingPlugins`
- Documentation de l'API de QGIS : `http://svn.qgis.org/api_doc/html/`
- Documentation de Qt : `http://doc.trolltech.com/4.3/index.html`

15 Écrire une extension en Python pour QGIS

Dans cette section vous trouverez un cours pour débutant pour écrire des extensions Python Pour QGIS. Il est basé sur le workshop "Étendre les fonctionnalités de QGIS avec des extensions en Python" réalisé lors du FOSS4G 2008 par Dr. Marco Hugentobler, Dr. Horst Düster et Tim Sutton.

En plus d'écrire des extensions en python pour QGIS, il est également possible d'utiliser PyQGIS dans une console de ligne de commande python qui est principalement utilisée pour déboguer ou pour écrire des applications indépendantes en Python avec leurs propres interfaces en utilisant la bibliothèque principale de QGIS.

15.1 Pourquoi Python et à propos de la licence

Python est un langage de script qui a été conçu afin d'être facile à écrire. Il a un mécanisme qui nettoie automatiquement la mémoire qui n'est plus utilisée (collecteur de déchet). Un avantage supplémentaire est que plusieurs programmes écrits en C++ ou Java offrent la possibilité d'écrire des extensions en Python, comme OpenOffice.org ou GIMP. C'est donc un bon investissement d'apprendre le langage Python.

Les extensions PyGQIS utilisent les fonctionnalités de libqgis_core.so et libqgis_gui.so. Comme les deux sont publiés sous licence GPL, les extensions Python pour QGIS doivent être publiés sous licence GPL également. Cela signifie que vous pouvez utiliser votre extension dans n'importe quel but et vous n'êtes pas obligé de les publier. Toutefois, si vous voulez les publier, ils doivent l'être dans les conditions de la licence GPL.

15.2 ce que vous avez besoin d'installer pour démarrer

Sur les ordinateurs du labs, tout ce qui est nécessaire est déjà installé. Si vous programmez chez vous, vous aurez besoin des bibliothèques et programmes suivants :

– QGIS ;
– Python ;
– Qt ;
– PyQT ;
– Outils de développement PyQt.

Si vous utilisez un système Linux ou équivalent, il existe des binaires pour toutes les distributions majeures. Pour les utilisateurs de Windows, l'installateur PyQT contient déjà Qt, PyQT et les outils de développement de PyQT.

15.3 Programmer une extension PyQGIS en quatre étapes

Notre exemple d'extension restera intentionnellement simple. Il ajoute un bouton à la barre de menu de QGIS. Si le bouton est cliqué, une boîte de dialogue apparait dans laquelle un utilisateur peut charger un fichier shape.

Pour chaque extension Python, un répertoire dédié qui contient les fichiers du extensions est nécessaire. Par défaut, QGIS cherche des extensions dans $QGIS_DIR/share/qgis/python/plugins et $HOME/.qgis/python/plugins. Remarquez que les extensions installés dans ce dernier sont seulement visible par l'utilisateur.

Étape 1 : reconnaissance d'une extension par le gestionnaire d'extension

Chaque extension Python est contenu dans son propre répertoire. Lors de démarrage de QGIS celui-ci parcourra chaque sous-répertoire spécifique au système et initialisera toutes les extensions qu'il trouvera.

- Linux et autre UNIX :
  ./share/qgis/python/plugins
  /home/$USERNAME/.qgis/python/plugins
- Mac OS X :
  ./Contents/MacOS/share/qgis/python/plugins
  /Users/$USERNAME/.qgis/python/plugins
- Windows :
  C :\Program Files\QGIS\python\plugins
  C :\Documents and Settings\$USERNAME\.qgis\python\plugins

Une fois réalisé, le plugin s'affichera dans gestionnaire de plugin...

Astuce 44 DEUX RÉPERTOIRES DE PLUGINS PYTHON

Il y a deux répertoires contenant les extensions en python. $QGIS_DIR/share/qgis/python/plugins a été conçu principalement pour les extensions principales tandis que $HOME/.qgis/python/plugins pour les extensions seulement visibles par l'utilisateur, mais aussi masque les extensions principales de même nom, ce qui peut être pratique pour les mettre à jour.

Pour fournir les informations nécessaires pour QGIS, le plugin nécessite d'implémenter les méthodes *name()*, *description()* et *version()* qui renvoient les chaînes descriptives. *qgisMinimumVersion()* doit renvoyer une forme simple, par exemple "1.0". Une extension nécessite également une méthode *classFactory(QgisInterface)* qui est appelée par le gestionnaire d'extension pour créer une instance de l'extension. L'argument de type QGisInterface est utilisé par l'extension pour accéder aux fonctions de l'instance QGIS. Nous allons travailler avec cet objet à l'étape 2.

Notez que, contrairement aux autres langages de programmation, l'indentation est très importante. L'interpréteur Python renvoie une erreur si elle n'est pas correcte.

Pour nos plugins nous créons un répertoire de plugin 'foss4g_plugin' dans `$HOME/.qgis/python/plugins`. Puis nous ajoutons deux nouveaux fichiers textes dans ce répertoire `foss4gplugin.py` et `__init__.py`.

Le fichier `foss4gplugin.py` contient la classe de l'extension :

```
# -*- coding: utf-8 -*-
# Import des bibliothèques PyQt et QGIS
from PyQt4.QtCore import *
from PyQt4.QtGui import *
from qgis.core import *
# Initialisation des ressources Qt à partir du fichier resources.py
import resources

class FOSS4GPlugin:

def __init__(self, iface):
# Sauve la référence à l'interface QGIS
  self.iface = iface

def initGui(self):
  print 'Initialising GUI'

def unload(self):
  print 'Unloading plugin'
```

Le fichier `__init__.py` contient les méthodes *name()*, *description()*, *version()*, *qgisMinimumVersion()* et *authorName()* évoqués plus haut. Comme nous somme en train de créer une nouvelle instance de la classe plugin (extension)s, nous devons importer le code de cette classe :

```
# -*- coding: utf-8 -*-
from foss4gplugin import FOSS4GPlugin
def name():
  return "FOSS4G example"
def description():
  return "A simple example plugin to load shapefiles"
def version():
  return "0.1"
def qgisMinimumVersion():
  return "1.0"
def authorName():
  return "John Developer"
```

```
def classFactory(iface):
  return FOSS4GPlugin(iface)
```

Maintenant l'extension possède l'infrastructure nécessaire pour apparaître dans le gestionnaire d'extension QGIS et être chargé/déchargé.

Étape 2 : Créer un icône pour le plugin

Pour que votre icône graphique soit disponible dans votre programme, nous avons besoin d'un fichier ressource. Dans ce fichier ressource, le graphique est contenu sous forme hexadécimale. Heureusement, nous n'avons pas à nous occuper de sa représentation parce que nous utilisons le compilateur pyrcc, un outil qui lit le fichier `resources.qrc` et créé un fichier ressource.

Le fichier `foss4g.png` et le ficher `resources.qrc` peuvent être téléchargé à partir de `http://karlinapp.ethz.ch/python_foss4g`. Déplacez ces fichiers dans le répertoire de l'extension exemple `$HOME/.qgis/python/plugins/foss4g_plugin` et entrez : pyrcc4 -o ressources.py ressources.qrc.

Étape 3 : ajouter un bouton au menu

Dans cette partie, nous allons implémenter le contenu des méthodes *initGui()* et *unload()*. Nous avons besoins d'une instance de la classe **QAction** qui exécute la méthode *run()* de l'extension. Avec l'objet action, nous somme alors capable de générer l'entrée du menu et le bouton :

```
import resources

  def initGui(self):
    # Créer une action qui démmarera la configuration du plugin
    self.action = QAction(QIcon(":/plugins/foss4g_plugin/foss4g.png"), "FOSS4G
plugin",self.iface.getMainWindow())
    # Connecter l'action à la méthode run
    QObject.connect(self.action, SIGNAL("activated()"), self.run)

    # Ajoutez le bouton de la barre d'outil et l'entrée du menu
    self.iface.addToolBarIcon(self.action)
    self.iface.addPluginMenu("FOSS-GIS plugin...", self.action)

    def unload(self):
    # Supprime les entrées des menu et de l'icône
    self.iface.removePluginMenu("FOSSGIS Plugin...", self.action)
    self.iface.removeToolBarIcon(self.action)
```

Étape 4 : charger une couche à partir d'un shapefile

Dans cette étape nous allons implémenter les fonctionnalités réelles de l'extension dans la méthode methodrun() La méthode Qt4 *QFileDialog : :getOpenFileName* ouvre une boîte de dialogue et renvoie le chemin du fichier choisit. Si l'utilisateur annule la boîte de dialogue, le chemin est un objet null, que nous allons tester. Puis nous appelons la méthode *addVectorLayer* de l'objet interface qui charge la couche. La méthode possède seulement trois arguments : le chemin du fichier, le nom de la couche qui sera affichée dans la légende et le nom du fournisseur de données. Pour les Shapefiles, c'est 'ogr' car QGIS utilise en interne la bibliothèque OGR pour accéder aux shapefiles :

```
def run(self):
fileName = QFileDialog.getOpenFileName(None,QString.fromLocal8Bit("Select a file:"),
"", "*.shp *.gml")
if fileName.isNull():
  QMessageBox.information(None, "Cancel", "File selection canceled")
  else:
  vlayer = self.iface.addVectorLayer(fileName, "myLayer", "ogr")
```

15.4 Comiter l'extension dans un dépôt

Si vous avez écrit une extension vous pouvez trouver utile et vouloir le partager avec d'autres utilisateurs, vous êtes invité à le télécharger sur le dépôt de contribution des utilisateurs de QGIS.

- Préparer un répertoire d'extensions contenant les fichiers nécessaires (assurez-vous qu'il n'y ait pas de fichiers .pyc compilés, de répertoires .svn de subversion, etc.)
- Faîtes une archive zip, incluant le répertoire. Assurez-vous que le nom du fichier zip est exactement le même que le répertoire à l'intérieur (sauf bien sur avec l'extension .zip). Sinon l'installateur d'extension ne sera pas capable de relier l'extension disponible avec celui installé localement.
- Téléchargez le dans le dépôt : `http://pyqgis.org/admin/contributed` (vous devez vous enregistrer la première fois). S'il vous plait, ayez une attention particulière lors du remplissage du formulaire. Spécialement le champ du numéro de version qui est souvent remplie incorrectement ce qui pose problème à l'installateur d'extension et cause de fausse notification de mise à jour disponible.

15.5 Plus d'informations

Comme vous pouvez voir, vous avez besoin d'informations de différentes sources pour écrire des extensions PyQGIS. Les développeurs de plugins doivent connaître Python et l'interface d'extension de QGIS ainsi que les classes et outils de Qt4. Au début il est important d'apprendre à partir des exemples et de copier les mécanismes d'extensions existants. En utilisant l'installateur d'extensionns

de QGIS, qui est lui même une extension Python, il est possible de télécharger plusieurs extensions et de les étudier.

Il y a de nombreuses documentations qui peuvent être utile pour les programmeurs de PyQGIS :

– wiki de QGIS : `http://wiki.qgis.org/qgiswiki/PythonBindings`
– Documentation de l'API QGIS : `http://doc.qgis.org/index.html`
– Documentation Qt : `http://doc.trolltech.com/4.3/index.html`
– PyQt : `http://www.riverbankcomputing.co.uk/pyqt/`
– Cours sur Python : `http://docs.python.org/`
– Un livre sur les SIG bureautiques et QGIS. Il contient un chapitre sur la programmation d'extension PyQGIS : `http://www.pragprog.com/titles/gsdgis/desktop-gis`

Vous pouvez également écrire des extensions pour QGIS en C++. Lisez la section 14 pour plus d'information là dessus.

16 Créer des applications en C++

Pas tout le monde désire une application SIG complète. Pafois vous voulez juste un widget inclus dans votre application qui affiche une carte alors que le but principal de l'application porte sur autre chose. Peut-être une interface cliente d'une base de données avec une carte ? Cette section fournit deux exemples simples de code écrit par Tim Sutton. Ils sont disponibles dans le dépôt de code subversion de QGIS réunis avec d'autres tutoriels intéressants. Récupérez le dépôt complet à partir de `https ://svn.osgeo.org/qgis/trunk/code_examples/`

16.1 Créer un simple widget de cartographie

À travers ce petit tutoriel nous allons faire un petit tour dans la création de widget simple de cartographie. Mais il vous donnera une idée du potentiel de l'utilisation de QGIS comme composant de cartographie encapsulé. Avant de commencer, un grand merci à Francis Bolduc qui a écrit le début de cette démo. Il a gracieusement accepté de rendre son travail disponible.

Nous commençons en ajoutant les inclusions nécessaires pour notre application :

```
//
// Inclusions pour QGIS
//
#include <qgsapplication.h>
#include <qgsproviderregistry.h>
#include <qgssinglesymbolrenderer.h>
#include <qgsmaplayerregistry.h>
#include <qgsvectorlayer.h>
#include <qgsmapcanvas.h>
//
// Inclusions pour Qt
//
#include <QString>
#include <QApplication>
#include <QWidget>
```

Nous utilisons QgsApplication au lieu de QApplication de Qt et nous obtenons des bénéfices supplémentaires de différentes méthodes statiques qui peuvent être utilisée pour localiser les chemins des bibliothèques, etc.

Le registre des fournisseurs est un singleton qui garde la trace des extensions des fournisseurs de données vecteurs. Il réalise tout le travail de chargement des extensions, etc. à votre place. Le moteur de rendu de symbole simple est la classe de symbologie la plus simple. Il réalise un rendu de points,

lignes ou polygones dans une seule couleur qui est choisie aléatoirement par défaut (bien que vous pouvez le définir vous même). Chaque couche vecteur doit avoir une sémiologie qui lui est associée.

Le registre de couche de la carte garde la trace de toutes les couches que vous utilisez. La classe de la couche vecteur hérite de maplayer et l'étend pour inclure des fonctionnalités spécialisées pour les données vecteurs.

Enfin, le canevas de la carte est vraiment la partie la plus importante, car c'est le widget dans lequel notre carte sera dessinée.

Maintenant nous pouvons nous initialiser notre application ...

```
int main(int argc, char ** argv)
{
  // Début de l'application
  QgsApplication app(argc, argv, true);

  QString myPluginsDir        = "/home/timlinux/apps/lib/qgis";
  QString myLayerPath         = "/home/timlinux/gisdata/brazil/BR_Cidades/";
  QString myLayerBaseName     = "Brasil_Cap";
  QString myProviderName      = "ogr";
```

Nous avons donc maintenant une qgsapplication et définie des variables. Puisque j'ai testé cela sur Ubuntu 8.10, j'ai défini la localisation de l'extension fournisseur de vecteur comme étant dans mon répertoire d'installation pour le développement. Cela aurait plus de sens, de garder la bibliothèque de QGIS dans un des chemins de recherche standard de la bibliothèque sur votre système (par exemple /usr/lib/) mais cette manière conviendra, pour le moment.

Les deux variables suivantes définies ici pointent vers le shapefile que je vais utiliser (et vous devez substituer vos propres données ici).

Le nom du fournisseur est important - il indique à QGIS quel fournisseur de données à utiliser pour charger le fichier. Habituellement, vous utiliserez 'ogr' ou 'postgres'.

Maintenant nous pouvons avancer sur la véritable création de notre objet couche.

```
  // Instentie le Registre de Fournisseur
  QgsProviderRegistry::instance(myPluginsDir);
```

D'abord, nous initialisons le registre de fournisseur. Comme c'est une classe simple, nous utilisons l'appel de l'instance statique et lui passons le chemin de recherche des bibliothèques du fournisseur. Lors de son initialisation, il scannera ce chemin pour les bibliothèques du fournisseur.

Maintenant nous créons une couche ...

```
  QgsVectorLayer * mypLayer =
      new QgsVectorLayer(myLayerPath, myLayerBaseName, myProviderName);
  QgsSingleSymbolRenderer *mypRenderer = new
QgsSingleSymbolRenderer(mypLayer->geometryType());
  QList <QgsMapCanvasLayer> myLayerSet;

  mypLayer->setRenderer(mypRenderer);
  if (mypLayer->isValid())
  {
    qDebug("Layer is valid");
  }
  else
  {
    qDebug("Layer is NOT valid");
  }

  // Ajout de la couche au registre de couche
  QgsMapLayerRegistry::instance()->addMapLayer(mypLayer, TRUE);
  // Ajout de la couche à l'ensemble des couches
  myLayerSet.append(QgsMapCanvasLayer(mypLayer, TRUE));
```

Le code est assez explicite ici. Nous créons une couche en utilisant les variables que nous avons définies plus tôt. Puis nous assignons à la couche un moteur de rendu. Lorsque nous créons un moteur de rendu, nous avons besoins de définir le type de géométrie, ce qui est fait en demandant à la couche son type de géométrie. Puis nous ajoutons la couche à un ensemble de couches (qui est utilisé par QgsMapCanvas pour garder en mémoire quelles couches doivent être affichées et dans quel ordre) et au registre maplayer. Enfin, nous nous assurons que la couche sera visible.

Maintenant nous créons un cadre de carte dans lequel nous pouvons dessiner la couche.

```
  // Creer le cadre de la carte
  QgsMapCanvas * mypMapCanvas = new QgsMapCanvas(0, 0);
  mypMapCanvas->setExtent(mypLayer->extent());
  mypMapCanvas->enableAntiAliasing(true);
  mypMapCanvas->setCanvasColor(QColor(255, 255, 255));
  mypMapCanvas->freeze(false);
  // Definir l'ensemble des couches du cadre de carte
  mypMapCanvas->setLayerSet(myLayerSet);
  mypMapCanvas->setVisible(true);
```

```
  mypMapCanvas->refresh();
```

Une fois encore il n'y a rien de particulièrement difficile ici. Nous créons le cadre et nous lui définissons son étendue à celui de la couche. Puis nous modifions légèrement le cadre pour dessiner les vecteurs antialiésés. Enfin, nous définissons la couleur d'arrière-plan, débloquons le cadre, le rendons visible et le rafraichissons.

```
  // Debut de la boucle d'evenement de l'application
  return app.exec();
}
```

À la dernière étape, nous démarrons simplement la boucle événementielle de Qt et c'est terminé. Vous pouvez vérifier, compiler et lancer cet exemple en utilisant cmake comme ceci :

```
svn co
https://svn.osgeo.org/qgis/trunk/code_examples/1_hello_world_qgis_style
cd 1_hello_world_qgis_style
mkdir build
#en option specifiez où QGIS est installe (doit fonctionner sur toutes les plateformes)
#si QGIS est installe dans /usr ou /usr/local vous pouvez laisser tomber l'etape suivante
export LIB_DIR=/home/timlinux/apps
cmake ..
make
./timtut1
```

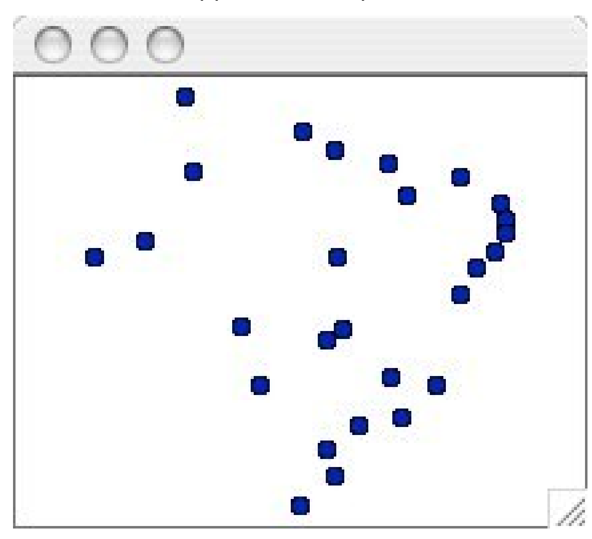
Lorque nous le compilons et le lançons voici ce à quoi ressemble l'application lancée :

16.2 Utiliser QgsMapCanvas

Dans la section 16.1 nous vous avons montré l'utilisation de l'api QgsMapCanvas pour créer une application simple qui charge un shapefile et affiche les points. Mais à quoi sert une carte si vous ne pouvez pas interagir avec ?

Dans cette seconde partie, j'étendrais la première en faisant une application QMainWindow avec un menu, une barre d'outils et une zone de carte. Nous vous avons montré comment utiliser QgsMapTool - la classe de base pour tous les outils qui nécessite d'interagir avec le cadre de carte. Le but est de fournir un projet de démonstration, je ne promets donc pas d'écrire le code C++ le plus robuste et le plus élégant. Le projet fournira 4 icônes de barres d'outils.

FIG. 58: Application simple en C++ X



- charger une carte (le nom de la couche est codé en dur dans l'application)
- dézoomer
- zoomer
- déplacer

Dans le répertoire de travail du code de ce cours, vous trouverez quelques fichiers incluant les sources c++, des icônes et un fichier de données simple dans le répertoire data. Il y a aussi des fichiers .ui pour la fenêtre principale.

Note : vous aurez besoin d'éditer lefichier .pro dans le répertoire svn ci-dessus pour qu'il corresponde à votre système.

Puisqu'une grande partie du code est le même que la partie précédente, nous nous focaliserons sur les spécificités de MapTool - le reste des détails de l'implémentation peut être regardé en récupérant le projet à partir du SVN. Un QgsMapTool est une classe qui interagit avec le MapCanvas en utilisant le pointeur de la souris. QGIs a un certain nombre de QgsMapTools implémenté, et vous pouvez sous-classer QgsMapTool pour créer les vôtres. Dans le fichier mainwindow.cpp vous verrez que j'ai inclus les en-têtes pour QgsMapTools près du début du fichier :

```
//
// QGIS Map tools
//
#include "qgsmaptoolpan.h"
#include "qgsmaptoolzoom.h"
//
// Il y a d'autres en tete pour les outils de la carte disponibles
```

```
// (non utilise dans cet exemple)
//
//#include "qgsmaptoolcapture.h"
//#include "qgsmaptoolidentify.h"
//#include "qgsmaptoolselect.h"
//#include "qgsmaptoolvertexedit.h"
//#include "qgsmeasure.h"
```

Comme vous pouvez le voir, je n'utilise que deux types de sous-classe MapTool pour cette partie, mais il y en a plus de disponibles dans la bibliothèque QGIS. Agrafer notre MapTools au cadre est très facile en utilisant le mécanisme normal slot/signal de Qt4.

```
//creer le comportement action
connect(mActionPan, SIGNAL(triggered()), this, SLOT(panMode()));
connect(mActionZoomIn, SIGNAL(triggered()), this, SLOT(zoomInMode()));
connect(mActionZoomOut, SIGNAL(triggered()), this, SLOT(zoomOutMode()));
connect(mActionAddLayer, SIGNAL(triggered()), this, SLOT(addLayer()));
```

Puis nous réalisons une barre d'outils pour contenir nos boutons. Notez que les actions mpAction* ont été créées dans QtDesigner.

```
//creer une petite barre d'outils
mpMapToolBar = addToolBar(tr("File"));
mpMapToolBar->addAction(mpActionAddLayer);
mpMapToolBar->addAction(mpActionZoomIn);
mpMapToolBar->addAction(mpActionZoomOut);
mpMapToolBar->addAction(mpActionPan);
```

Ceci également est une possibilité de Qt assez simple.Maintenant nous créons nos trois outils pour la carte :

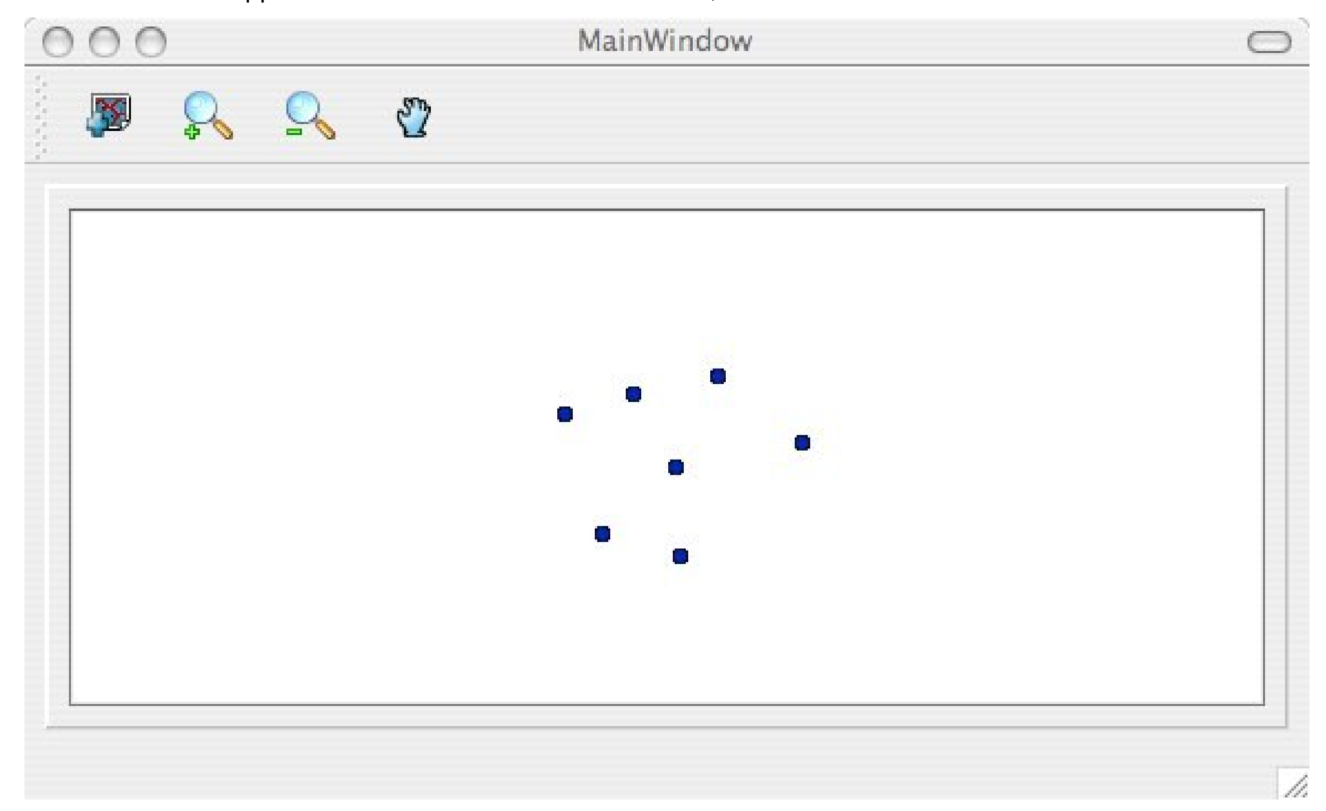
```
//creer les outils de le carte
mpPanTool = new QgsMapToolPan(mpMapCanvas);
mpPanTool->setAction(mpActionPan);
mpZoomInTool = new QgsMapToolZoom(mpMapCanvas, FALSE); // false = in
mpZoomInTool->setAction(mpActionZoomIn);
mpZoomOutTool = new QgsMapToolZoom(mpMapCanvas, TRUE ); //true = out
mpZoomOutTool->setAction(mpActionZoomOut);
```

Rien de bien compliqué ici aussi - nous créons des instances d'outils, chacune d'elles est associée avec le même mapcanvas, et une QAction différente. Quand l'utilisateur sélectionne une des icônes

de la barre d'outils, le MapTool actif pour le canevas est défini. Par exemple quand l'icône déplacement est cliquée, nous réalisons cela :

```
void MainWindow::panMode()
{
   mpMapCanvas->setMapTool(mpPanTool);
}
```

FIG. 59: Application QMainWindow avec un menu, une barre d'outils et une zone de carte X



Conclusion

Comme vous pouvez le voir, étendre notre exemple précédent en quelque chose de plus fonctionnel en utilisant MapTools est vraiment facile et nécessite seulement quelques lignes de codes pour chaque MapTool que vous voulez fournir.

Vous pouvez récupérer et compiler ce tutoriel en utilisant SVN et CMake avec les étapes suivantes :

```
svn co https://svn.osgeo.org/qgis/trunk/code_examples/2_basic_main_window
cd 2_basic_main_window
```

```
mkdir build
#en option spécifiez où QGIS est installe (doit fonctionner sur toutes les plateformes)
#si QGIS est installe dans /usr ou /usr/local vous pouvez laisser tomber l'étape suivante
export LIB_DIR=/home/timlinux/apps
cmake ..
make
./timtut2
```

17 Créer des applications PyQGIS

Un des objectifs de QGIS est de fournir non seulement une application mais également un jeu de bibliothèques pouvant être utilisées pour créer de nouvelles applications. Cet objectif a été rempli par le reformatage des bibliothèques qui a eut lieu après la sortie de la version 0.8. Depuis la version 0.9, le développement d'applications autonomes est possible en utilisant les langages C++ ou Python. Nous vous recommandons d'utiliser QGIS 1.0.0 ou une version plus récente comme base pour vos applications Python car c'est depuis cette version que nous proposons une API stable et cohérente.

Dans ce chapitre nous allons décrire rapidement le processus de création d'une application Python autonome. Le blog QGIS propose quelques exemples de création d'applications PyQGIS [13].

Les fonctionnalités que nous souhaitons avoir dans cette application sont :

– Charger une couche vecteur
– Déplacements panoramiques
– Zoom + et -
– Zoom sur l'étendue globale de la couche
– Choix des couleurs au chargement de la couche

Il s'agit ici des fonctionnalités minimales et relativement essentielles. Commençons par créer l'interface en utilisant Qt Designer.

17.1 Design de l'interface

Comme nous créons une application minimaliste, nous allons avoir la même approche pour l'interface. En nous servant de Qt Designer, nous créons une simple fenêtre principale (*MainWindow*) sans menu ni barre d'outils. Nous aurons une fenêtre vide sur laquelle nous allons pouvoir travailler. Pour créer la fenêtre principale :

1. Créez un répertoire pour le développement de l'application et définissez-le comme répertoire de travail
2. Lancez Qt Designer
3. La boîte de dialogue New Form devrait apparaître. Si ce n'est pas le cas, choisissez New Form... dans le menu File.
4. Choisissez Main Window dans la liste templates/forms
5. Cliquez sur Create
6. Redimensionnez la nouvelle fenêtre à une taille convenable

[13] Une application créée en utilisant Python et la mise à disposition par encapsulation (*bindings*) des bibliothèques de QGIS

7. Trouvez le widget Frame dans la liste (dans Containers) et faites-le glisser dans la fenêtre principale précédemment créée.
8. Cliquez en dehors du “frame” pour sélectionner la fenêtre principale
9. Cliquez sur l’outil Lay Out in a Grid. Le “frame” vient alors remplir toute la fenêtre principale.
10. Sauvegardez le formulaire sous `mainwindow.ui`
11. Sortez de Qt Designer : Exit

Compilez ensuite le formulaire en utilisant le compilateur d’interface PyQt :

```
pyuic4 -o mainwindow_ui.py mainwindow.ui
```

Ceci permet de créer le code source Python pour l’interface constituée de la fenêtre principale. Ensuite nous devons créer le code de l’application qui ajoutera à cette fenêtre vide quelques outils que nous pourrions utiliser.

17.2 Création de la fenêtre principale

Nous pouvons maintenant écrire la classe **MainWindow** qui effectuera effectivement le travail. Comme cela nécessite un grand nombre de lignes de code, nous allons les décrire par morceaux en commençant par la section import et la préparation de l’environnement :

```
# Loosely based on:
#   Original C++ Tutorial 2 by Tim Sutton
#   ported to Python by Martin Dobias
#   with enhancements by Gary Sherman for FOSS4G2007
# Licensed under the terms of GNU GPL 2

from PyQt4.QtCore import *
from PyQt4.QtGui import *
from qgis.core import *
from qgis.gui import *
import sys
import os
# Import our GUI
from mainwindow_ui import Ui_MainWindow

# Environment variable QGISHOME must be set to the 1.0 install directory
# before running this application
qgis_prefix = os.getenv("QGISHOME")
```

Certaines lignes doivent vous sembler familières de nos extensions, notamment les importations PyQt4 et QGIS. Les spécificités à noter sont les importations de notre IHM ligne 14 et de notre bibliothèque CORE ligne 9.

Notre application a besoin de savoir où se trouve l'installation de QGIS. C'est pour cette raison que nous définissons la variable d'environnement `QGISHOME` pour pointer sur le répertoire d'installation de QGIS 1.x. à la ligne 18, nous stockons cette valeur pour une utilisation ultérieure.

Ensuite, nous devons créer notre classe **MainWindow** qui déterminera la logique de fonctionnement de notre application.

```
21 class MainWindow(QMainWindow, Ui_MainWindow):
22
23   def __init__(self):
24     QMainWindow.__init__(self)
25
26     # Required by Qt4 to initialize the UI
27     self.setupUi(self)
28
29     # Set the title for the app
30     self.setWindowTitle("QGIS Demo App")
31
32     # Create the map canvas
33     self.canvas = QgsMapCanvas()
34     # Set the background color to light blue something
35     self.canvas.setCanvasColor(QColor(200,200,255))
36     self.canvas.enableAntiAliasing(True)
37     self.canvas.useQImageToRender(False)
38     self.canvas.show()
39
40     # Lay our widgets out in the main window using a
41     # vertical box layout
42     self.layout = QVBoxLayout(self.frame)
43     self.layout.addWidget(self.canvas)
44
45     # Create the actions for our tools and connect each to the appropriate
46     # method
47     self.actionAddLayer = QAction(QIcon("(qgis_prefix + "/share/qgis/themes/classic/mActior
48     \
49         "Add Layer", self.frame)
50     self.connect(self.actionAddLayer, SIGNAL("activated()"), self.addLayer)
51     self.actionZoomIn = QAction(QIcon("(qgis_prefix + "/share/qgis/themes/classic/mActionZc
52         "Zoom In", self.frame)
```

```
53      self.connect(self.actionZoomIn, SIGNAL("activated()"), self.zoomIn)
54      self.actionZoomOut = QAction(QIcon("(qgis_prefix + "/share/qgis/themes/classic/mActionZo
55          "Zoom Out", self.frame)
56      self.connect(self.actionZoomOut, SIGNAL("activated()"), self.zoomOut)
57      self.actionPan = QAction(QIcon("(qgis_prefix + "/share/qgis/themes/classic/mActionPan.pn
58          "Pan", self.frame)
59      self.connect(self.actionPan, SIGNAL("activated()"), self.pan)
60      self.actionZoomFull = QAction(QIcon("(qgis_prefix + "/share/qgis/themes/classic/mActionZ
61          "Zoom Full Extent", self.frame)
62      self.connect(self.actionZoomFull, SIGNAL("activated()"),
63      self.zoomFull)
64
65      # Create a toolbar
66      self.toolbar = self.addToolBar("Map")
67      # Add the actions to the toolbar
68      self.toolbar.addAction(self.actionAddLayer)
69      self.toolbar.addAction(self.actionZoomIn)
70      self.toolbar.addAction(self.actionZoomOut);
71      self.toolbar.addAction(self.actionPan);
72      self.toolbar.addAction(self.actionZoomFull);
73
74      # Create the map tools
75      self.toolPan = QgsMapToolPan(self.canvas)
76      self.toolZoomIn = QgsMapToolZoom(self.canvas, False) # false = in
77      self.toolZoomOut = QgsMapToolZoom(self.canvas, True) # true = out
```

Les lignes 21 à 27 constituent la déclaration et l'initialisation basiques de **MainWindow** et la l'installation de l'interface utilisateur par la méthode *setupUi*. Ceci est nécessaire pour toutes les applications.

Ensuite nous définissons le titre de l'application qui soit plus significatif que `MainWindow` (ligne 30). Une fois cela effectué, nous sommes prêts à terminer l'interface utilisateur. Lors de sa création dans Designer, nous l'avons laissé très simple — juste une fenêtre principale et un frame. Vous auriez pu ajouter un menu et une barre d'outils en utilisant Designer mais nous allons le faire avec Python.

Des lignes 33 à 38, nous créons le canevas de la carte, définissons la couleur de l'arrière plan à bleu clair et activons l'antialiasing. Nous précisons également de ne pas utiliser **QImage** pour le rendu (faites moi confiance) et enfin nous rendons le canevas visible en appelant la méthode *show*.

Ensuite nous indiquons qu'il faut utiliser un alignement vertical dans le cadre et, ligne 43, y ajoutons le canevas de la carte.

Des lignes 48 à 63, nous définissons les actions et les connexions pour les outils de la barre d'outils. Pour chaque outil, nous créons un **QAction** utilisant l'icône que nous avons défini dans le thème

classique de QGIS. Ensuite nous connectons le signal `activated` de l'outil à la méthode de notre classe qui va gérer l'action. Ceci est similaire à la manière dont nous définissons les choses dans l'exemple de l'extension.

Une fois que nous avons les actions et les connexions, nous avons besoin de les ajouter à la barre d'outils. Ligne 66 à 72, nous créons la barre d'outils et y ajoutons chacun des outils.

Enfin, nous créons les trois outils cartographiques pour l'application (ligne 75 à 77). Nous allons utiliser les outils cartographiques dans un instant, quand nous définirons les méthodes qui rendront notre application fonctionnelle. Jetons un oeil aux méthodes des outils cartographiques.

```
78  # Set the map tool to zoom in
79  def zoomIn(self):
80    self.canvas.setMapTool(self.toolZoomIn)
81
82  # Set the map tool to zoom out
83  def zoomOut(self):
84    self.canvas.setMapTool(self.toolZoomOut)
85
86  # Set the map tool to
87  def pan(self):
88   self.canvas.setMapTool(self.toolPan)
89
90  # Zoom to full extent of layer
91  def zoomFull(self):
92    self.canvas.zoomFullExtent()
```

Pour chaque outil, nous avons besoin d'une méthode qui correspond à la connexion que nous avons faite pour chaque action. Ligne 79 à 88, nous définissons une méthode pour chacun des trois outils qui interagissent avec la carte. Quand un outil est activé par un clic dans la barre d'outils, la méthode correspondante est appelée ce qui "dit" à la carte qu'il s'agit de l'outil activé. Cet outil activé décide de ce qu'il se passe quand l'utilisateur clique sur le canevas.

L'outil `zoom sur l'étendue complète` n'est pas un outil cartographique – il fait son travail sans nécessiter de clic sur la carte. Quand il est activé, la méthode *zoomFullExtent* du canevas de la carte est appelée (ligne 92). Ceci termine l'implémentation de tous nos outils excepté l'outil `Ajouter une couche`.

Regardons cela tout de suite :

```
93  # Add an OGR layer to the map
94  def addLayer(self):
95    file = QFileDialog.getOpenFileName(self, "Open Shapefile", ".", "Shapefiles
```

```
96      (*.shp)")
97      fileInfo = QFileInfo(file)
98
99      # Add the layer
100      layer = QgsVectorLayer(file, fileInfo.fileName(), "ogr")
101
102     if not layer.isValid():
103       return
104
105     # Change the color of the layer to gray
106     symbols = layer.renderer().symbols()
107     symbol = symbols[0]
108     symbol.setFillColor(QColor.fromRgb(192,192,192))
109
110     # Add layer to the registry
111     QgsMapLayerRegistry.instance().addMapLayer(layer);
112
113     # Set extent to the extent of our layer
114     self.canvas.setExtent(layer.extent())
115
116     # Set up the map canvas layer set
117     cl = QgsMapCanvasLayer(layer)
118     layers = [cl]
119     self.canvas.setLayerSet(layers)
```

Dans la méthode *addLayer*, nous utilisons un **QFileDialog** pour récupérer le nom du shapefile à charger. Ceci est fait ligne 96. Notez que nous spécifions un “filtre”de sorte que la boîte de dialogue ne montrera que les fichiers d'extension `.shp`.

Ensuite ligne 97 nous créons un objet **QFileInfo** avec le chemin du shapefile. Maintenant la couche est prête à être créée à la ligne 100. En utilisant l'objet **QFileInfo** pour récupérer le nom du fichier à partir du chemin, nous donnons le nom du fichier comme nom de la couche quand elle sera créée. Pour s'assurer que la couche est valide et ne va pas causer des problèmes au chargement, nous la vérifions ligne 102. Si elle est mauvaise, nous l'ignorons et ne l'ajoutons pas au canevas de la carte.

Normalement les couches sont ajoutées avec des couleurs aléatoires. Ici, nous souhaitons personnaliser les couleurs pour un affichage plus plaisant. De plus, nous savons que nous allons ajouter la couche `world_borders` à la carte et cela sera joli sur notre fond bleu. Pour changer la couleur, nous devons récupérer le symbole utilisé pour le rendu et l'utiliser pour définir une nouvelle couleur de remplissage. Ceci est fait ligne 106 à 108.

Tout ce qu'il reste à faire est d'ajouter la couche au registre et quelques autres tâches (ligne 111 à 119). Tout ceci est standard pour ajouter une couche et le résultat final correspond aux frontières du

monde sur un fond bleu clair. La dernière chose à faire est de définir l’étendue de la carte à celle de la couche, ce que vous ne souhaitez peut-être pas faire si vous voulez ajouter d’autres couches à votre application.

Ceci le coeur de l’application et termine la classe **MainWindow**.

17.3 Fin de l’application

```
def main(argv):
  # create Qt application
  app = QApplication(argv)

  # Initialize qgis libraries
  QgsApplication.setPrefixPath(qgis_prefix, True)
  QgsApplication.initQgis()

  # create main window
  wnd = MainWindow()
  # Move the app window to upper left
  wnd.move(100,100)
  wnd.show()

  # run!
  retval = app.exec_()

  # exit
  QgsApplication.exitQgis()
  sys.exit(retval)


if __name__ == "__main__":
  main(sys.argv)
```

17.4 Lancer l’application

Nous pouvons maintenant lancer notre application et voir ce qu’il se passe. Bien entendu, comme la plupart des développeurs, vous l’avez déjà testé régulièrement en la développant. Avant de lancer l’application, nous devons définir quelques variables d’environnement.



```
export LD_LIBRARY_PATH=$HOME/qgis/lib%$
export PYTHONPATH=$HOME/qgis/share/qgis/python
export QGISHOME=$HOME/qgis%$
```

```
set PATH=C:\qgis;%PATH%
set PYTHONPATH=C:\qgis\python
set QGISHOME=C:\qgis
```

Nous supposons que :
– QGIS est installé dans votre répertoire home dans `qgis`.
– QGIS est installé dans `C :\qgis`.
Pour ajouter la couche `world_borders`, cliquez sur l'outil `Ajouter une couche` et sélectionnez le répertoire de données. Sélectionnez le shapefile et cliquez sur [Ouvrir] pour l'ajouter à la carte.

Créer une application PyQGIS est vraiment facile. En moins de 150 lignes de code, nous avons une application qui permet de charger un shapefile et de naviguer sur la carte. Si vous jouez un peu avec la carte, vous vous apercevrez que certaines fonctionnalités intégrées du canevas marchent également, dont la molette de la souris, le déplacement en maintenant la barre (espace) et en bougeant la souris.

Des applications sophistiquées ont déjà été créées avec PyQGIS et beaucoup sont en travaux. C'est assez impressionnant d'autant plus que ces développements ont eu lieu avant la sortie officielle de QGIS 1.0.

Astuce 45 DOCUMENTATION SUR PYQGIS

Que vous écriviez une extension ou une application PyQGIS, vous allez avoir besoin de vous référer à la fois à la documentation de l'API QGIS (`http://doc.qgis.org`) et au Guide des PyQt Python Bindings Reference (`http://www.riverbankcomputing.com/Docs/PyQt4/pyqt4ref.html`). Ces documents fournissent les informations nécessaires sur les classes et les méthodes que vous allez utiliser pour donner vie à vos créations Python.

18 Aide et support

18.1 Mailinglists

QGIS est en cours de développement par conséquent il ne fonctionne pas toujours comme attendu. La manière préférée d'obtenir de l'aide est de rejoindre la liste de diffusion qgis-users.

qgis-users

Vos questions atteindront une audience plus large et les réponses bénéficieront à tous. Vous pouvez rejoindre la liste de diffusion qgis-users en allant sur la page suivante :
`http://lists.osgeo.org/mailman/listinfo/qgis-user`

qgis-developer

Si vous êtes un développeur et que vous faites face à un problème plus technique, il est préfŕable de rejoindre la liste de diffusion qgis-developer :
`http://lists.osgeo.org/mailman/listinfo/qgis-developer`

qgis-commit

À chaque fois qu'un commit est réalisé sur le dépôt du code de QGIS un email est envoyé à cette liste. Si vous voulez être à jour de chaque changement au code en cours, vous pouvez vous inscrire à cette liste :
`http://lists.osgeo.org/mailman/listinfo/qgis-commit`

qgis-trac

Cette liste fournit une notification par mail liée à la gestion du projet, incluant les rapports de bugs, tâches, et demandes de fonctionnalités. Vous pouvez vous inscrire à cette liste ici :
`http://lists.osgeo.org/mailman/listinfo/qgis-trac`

qgis-community-team

Cette liste reçoit les mails des thématiques lié à la documentation, au contexte d'aide, au guide utilisateur, à ce qui est lié à Internet donc les sites, listes de diffusion, forums et efforts de traduction. Si vous voulez travailler sur le guide utilisateur, cette liste est un bon point de départ pour poser vos questions. Vous pouvez vous inscrire à cette liste ici :
`http://lists.osgeo.org/mailman/listinfo/qgis-community-team`

qgis-release-team

Cette liste reçoit les mails des thématiques comme les procédures de publication de version, paquetage binaire pour différents systèmes et annonce des nouvelles versions à un monde plus large. Vous pouvez vous inscrire à cette liste ici :
`http://lists.osgeo.org/mailman/listinfo/qgis-release-team`

qgis-psc

Cette liste est utilisée pour discuter des problèmes du Comité de Pilotage lié à l'ensemble de la gestion et de la direction de Quantum GIS. Vous pouvez vous inscrire à cette liste ici :
`http://lists.osgeo.org/mailman/listinfo/qgis-psc`

Vous êtes invité à vous inscrire à ces listes. S'il vous plait, souvenez-vous de contribuer à la liste en répondant à des questions et en partageant vos expériences. Remarquez que les listes qgis-commit et qgis-trac ont été configurées pour notification seulement et n'acceptent pas de mail d'utilisateurs.

18.2 IRC

Nous maintenons une présence sur IRC - rejoignez-nous sur le canal #qgis sur `irc.freenode.net`. S'il vous plait, patientez pour obtenir une réponse puisque la plupart des personnes font autre chose et cela peut leur prendre un peu de temps pour remarquer votre question. Un support commercial pour QGIS est disponible. Regardez la page du site `http://qgis.org/content/view/90/91` pour plus d'informations.

Si vous ratez une discussion sur IRC, pas de problème ! Nous loguons toutes les discussions afin que vous puisiez facilement les suivre. Allez simplement sur `http://logs.qgis.org` et lisez les logs IRC.

18.3 BugTracker

Bien que la liste de diffusion utilisateur est utile pour des questions générales du type 'Comment je réalise xyz dans QGIS ?', vous pouvez vouloir nous avertir de bugs dans QGIS. Vous pouvez soumettre un rapport de bug en utilisant le tracker de bug sur `https://trac.osgeo.org/qgis/`. Lors de la création d'un ticket pour un bug, fournissez s'il vous plait une adresse mail valide où nous pouvons vous demander des informations supplémentaires.

Garder en mémoire que votre bug peut ne pas avoir la priorité à laquelle vous vous attendiez (cela dépendra de sa sévérité). Certains bugs peuvent nécessiter du travail supplémentaire de la part des développeurs pour y remédier et la personne compétente n'est pas forcément disponible.

Les demandes de fonctionnalité peuvent être soumises également en utilisant le même système de

ticket que pour les bugs. Assurez-vous de sélectionner le type `enhancement`.

Si vous avez trouvé un bug et l'avez corrigé vous même, vous pouvez aussi soumettre un patch. Encore, le superbe système de ticket Trac sur `https://trac.osgeo.org/qgis/` a également ce type. Sélectionnez `patch` dans le menu type. Un des développeurs le vérifiera et l'appliquera à QGIS.
Ne vous alarmez pas si votre correctif n'est pas appliqué directement - les développeurs peuvent être occupés sur d'autres commits.

18.4 Blog

La communauté QGIS tient également un weblog (BLOG) sur `http://blog.qgis.org` qui publie d'intéressants articles à la fois pour les utilisateurs et les développeurs. Vous êtes invités à contribuer au blog après vous être enregistrés.

18.5 Wiki

Enfin, nous maintenons un site web wiki sur `http://wiki.qgis.org` où vous pouvez trouver diverses informations utiles liées au développement de QGIS, plan des versions, liens vers les sites de téléchargement, astuces de traduction des messages, etc. Parcourez le, il y a des choses intéressantes.

A Formats de données gérés

A.1 Formats OGR gérés

Au moment de la rédaction de ce document, les formats suivants sont gérés par la bibliothèque OGR. Les formats dont on sait qu'ils fonctionnent dans QGIS sont indiqués en **gras**.

- **Arc/Info Binary Coverage**
- Comma Separated Value (.csv)
- DODS/OPeNDAP
- **ESRI Shapefile**
- FMEObjects Gateway
- GML
- IHO S-57 (ENC)
- **Mapinfo File**
- **Fichier Mapinfo**
- Microstation DGN
- OGDI Vectors
- ODBC
- Oracle Spatial
- PostgreSQL[14]
- **SDTS**
- SQLite
- UK .NTF
- U.S. Census TIGER/Line
- VRT - Virtual Datasource

A.2 Formats raster GDAL

Au moment de la rédaction de ce document, les formats suivants sont gérés par la bibliothèque GDAL. Notez que, pour différentes raisons, ces formats ne fonctionnent peut-être pas tous dans QGIS. Par exemple, certains nécessitent des bibliothèques externes commerciales. Seuls les formats qui ont été bien testés apparaîtront dans la liste des types de fichiers lors du chargement d'un raster dans QGIS. Les autres formats non testés peuvent être chargés en sélectionnant le filtre *All files (*)*. Les formats dont on sait qu'ils fonctionnent dans QGIS sont indiqués en **gras**.

- **Arc/Info ASCII Grid**
- **Arc/Info Binary Grid (.adf)**
- Microsoft Windows Device Independent Bitmap (.bmp)
- BSB Nautical Chart Format (.kap)

[14]QGIS implémente ses propres fonctions PostgreSQL. OGR doit être compilé sans la gestion de PostgreSQL

– VTP Binary Terrain Format (.bt)
– CEOS (Spot pour le moment)
– First Generation USGS DOQ (.doq)
– New Labelled USGS DOQ (.doq)
– Military Elevation Data (.dt0, .dt1)
– ERMapper Compressed Wavelets (.ecw)
– Raster ESRI .hdr
– Raster ENVI .hdr
– Envisat Image Product (.n1)
– EOSAT FAST Format
– FITS (.fits)
– Graphics Interchange Format (.gif)
– **Rasters GRASS**[15]
– **TIFF / GeoTIFF (.tif)**
– Hierarchical Data Format Release 4 (HDF4)
– **Erdas Imagine (.img)**
– Atlantis MFF2e
– MNT japonnais (.mem)
– **JPEG JFIF (.jpg)**
– JPEG2000 (.jp2, .j2k)
– JPEG2000 (.jp2, .j2k)
– NOAA Polar Orbiter Level 1b Data Set (AVHRR)
– Erdas 7.x .LAN et .GIS
– Raster In Memory
– Atlantis MFF
– Multi-resolution Seamless Image Database MrSID
– NITF
– NetCDF
– OGDI Bridge
– PCI .aux Labelled
– PCI Geomatics Database File
– Portable Network Graphics (.png)
– Netpbm (.ppm,.pgm)
– **USGS SDTS DEM (*CATD.DDF)**
– SAR CEOS
– **USGS ASCII DEM (.dem)**
– X11 Pixmap (.xpm)

[15]La gestion des rasters GRASS est faite par l'extension de fournisseur de données QGIS GRASS

B Modules de la boîte à outils de GRASS

La console GRASS dans la boîte à outils GRASS permet d'accéder à quasiment tous (plus de 300) les modules de GRASS au moyen de la ligne de commande. Pour offrir un environnement de travail plus ergonomique, à peu près 200 des modules et fonctionnalités de GRASS disponibles sont aussi disponibles par des boîtes de dialogue graphique.

B.1 Modules d'import et d'export de GRASS de la boîte à outils

Cette section liste toutes les boîtes de dialogue de la boîte à outils de GRASS pour importer et exporter des données dans une location et jeu de données préalablement sélectionnés dans GRASS.

B.2 Modules de convertion de type de données de la boîte à outils de GRASS

This Section lists all graphical dialogs in the GRASS Toolbox to convert raster to vector or vector to raster data in a currently selected GRASS location and mapset.

B.3 Modules de configuration de la projections et de la région de la boîte à outils de GRASS

Cette section liste tous les boîtes de dialogue dans la boîte à outils de GRASS pour gérer et modifier la région du jeu de données sélectionné et de configurer la projection.

TAB. 8: Boîte à outils GRASS : modules d'import de données

Modules d'import de données dans la boîte à outils de GRASS	
Nom du module	**Objectif**
r.in.arc	Convertit un fichier raster ascii ARC/INFO d'ESRI (GRID) en une couche raster (binaire)
r.in.ascii	Convertit un fichier raster texte ASCII en une couche raster (binaire)
r.in.aster	Georeferencement, rectification, et import d'image Terra-ASTER et des MNT relatif en utilisant gdalwarp
r.in.gdal	Importe un fichier raster géré par GDAL dans une couche raster binaire de GRASS
r.in.gdal.loc	Importe un fichier raster géré par GDAL dans une couche raster binaire de GRASS et créer une région lui correspondant
r.in.gridatb	Importe un fichier GRIDATB.FOR (TOPMODEL)dans une couche raster de GRASS
r.in.mat	Importe un fichier binaire MAT-File(v4) dans une couche raster GRASS
r.in.poly	Crée des couches raster à partir de fichiers ascii de données polygonales/linéairesdans le répertoire sélectionné
r.in.srtm	Importe des fichiers SRTM HGT dans GRASS
i.in.spotvgt	Importe de fichier NDVI VGT de SPOT dans une couche raster
v.in.dxf	Importe une couche vecteur DXF
v.in.e00	Importe un fichier ESRI E00 dans une couche vecteur
v.in.garmin	Importe un vecteur à partir d'un GPS en utilisant gpstrans
v.in.gpsbabel	Importe un vecteur à partir d'un GPS en utilisant gpsbabel
v.in.mapgen	Importe des vecteurs MapGen ou MatLab dans GRASS
v.in.ogr	Importe des couches vectorieles OGR/PostGIS
v.in.ogr.loc	Importe des couches vectorielles OGR/PostGIS et créer une région leurs correspondants
v.in.ogr.all	Importe toutes les couches vectorielles OGR/PostGIS dans une source de données définit
v.in.ogr.all.loc	Importe toutes les couches vectorielles OGR/PostGIS dans une source de données définit et créer une région leurs correspondant

TAB. 9: Boîte à outils GRASS : modules d'export de données

Modules d'export des données dans la boîte à outils de GRASS	
Nom du module	**Objectif**
r.out.gdal.gtiff	Exporte une couche raster en Geo TIFF
r.out.arc	Convertit une couche raster dans un fichier ARCGRID d'ESRI
r.gridatb	Exporte une couche raster GRASS en fichier GRIDATB.FOR (TOPMO-DEL)
r.out.mat	Exporte un raster GRASS en un fichier binaire MAT-File
r.out.bin	Exporte un raster GRASS en tableau binaire
r.out.png	Exporte un raster GRASS dans une image non géoréférencé au format PNG
r.out.ppm	Convertit une couche raster GRASS dans un fichier image PPM à la résolution du pixel de CURRENTLY DEFINED REGION
r.out.ppm3	Convertit 3 couches raster GRASS (R,G,B) dans un fichier image PPM à la résolution du pixel CURRENTLY DEFINED REGION
r.out.pov	Convertit une couche raster dans un fichier avec un champ poids pour POVRAY
r.out.tiff	Exporte une couche raster GRASS dans une image TIFF de 8/24bit à la résolution du pixel de la région sélectionnée
r.out.vrml	Exporte une couche raster dans le format Virtual Reality Modeling Language (VRML)
v.out.ogr	Exporte une couche vecteur dans différents formats (bibliothèque OGR)
v.out.ogr.gml	Exporte une couche vectorielle en GML
v.out.ogr.postgis	Exporte une couche vectorielle en différents formats (bibliothèque OGR)
v.out.ogr.mapinfo	Export au format Mapinfo d'une couche vectorielle
v.out.ascii	Convertit une couche vecteur binaire de GRASS en une couche vectorielle ASCII de GRASS
v.out.dxf	Convertit un vecteur de GRASS en DXF

TAB. 10: boîte à outils de GRASS : modules de conversion detype de données

Modules de conversion de types de données dans la boîte à outils de GRASS	
Nom du module	**Objectif**
r.to.vect.point	Convertit un raster en points vectoriels
r.to.vect.line	Convertit un raster en lignes vectorielles
r.to.vect.area	Convertit un raster en polygones vectoriels
v.to.rast.constant	Convertit un vecteur en raster en utilisant une constante
v.to.rast.attr	Convertit un vecteur en raster en utilisant des valeurs attributaires

TAB. 11: Boîte à outils de GRASS : modules de configuration de la projection et de la région

Modules de configuration de la projection et de la région de la boîte à outils de GRASS	
Nom du module	**Objectif**
g.region.save	Sauve la région actuelle dans la région nommée
g.region.zoom	Réduction de la région courante jusqu'à ce qu'il renvoie des données non-NULL à partir d'une carte raster
g.region.multiple.raster	Défini la région pour correspondre à de multiples couches raster
g.region.multiple.vector	Définis la région pour correspondre à de multiples couches vecteur
g.proj.print	Affiche des informations de la projection de la localisation actuelle
g.proj.geo	Affiche des informations de la projection à partir d'un fichier géoréférencé (raster, vecteur ou image)
g.proj.ascii.new	Affiche des informations de la projection à partir d'un fichier ASCII géoréférencé contenant une description WKT de la projection
g.proj.proj	Affiche des informations de la projection à partir d'un fichier de description de la projection PROJ.4
g.proj.ascii.new	Affiche des informations de la projection à partir d'un fichier ASCII géoréférencé contenant une description WKT de la projection et créé une nouvelle location basée sur celui-ci
g.proj.geo.new	Affiche des informations de la projection à partir d'un fichier géoréférencé (raster, vecteur ou image) et créé une nouvelle location basé sur celui-ci
g.proj.proj.new	Affiche des informations de la projection à partir d'un fichier de description de la projection PROJ.4 et créé une nouvelle location basée sur celui-ci
m.cogo	Une commande simple pour convertir des mesures de distances et d'orientation en coordonnées et vice-versa. Il suppose un système de coordonnés cartésiennes

B.4 Modules de données raster de la boîte à outils de GRASS

Cette section liste toutes les boîtes de dialogue dans la boîte à outils de GRASS pour utiliser et analyser des données raster dans un jeu de données et une région de GRASS sélectionnés.

TAB. 12: Boîte à outils de GRASS : Modules de développements de couches raster

Modules de développements de couches raster de la boîte à outils de GRASS	
Nom du module	**Objectif**
r.compress	Compresse et décompresse des couches raster
r.region.region	Définis la définition des frontières à la région par défaut ou celle actuelle
r.region.raster	Définis la définition des frontières à partir d'une couche raster existante
r.region.vector	Définis la définition des frontières à partir d'une couche vecteur existante
r.region.edge	Définis la définition des frontières par le bord (n-s-e-o)
r.region.alignTo	Définis la région sur laquelle aligner la couche raster
r.null.val	Transforme les cellules avec des valeurs en cellules nulles
r.null.to	Transforme les cellules nulles en cellules avec une valeur
r.quant	Cette routine produit le fichier de quantification pour une carte en virgule flottante
r.resamp.stats	Reéchantillonne des couches raster en utilisant l'aggrégation
r.resamp.interp	Reéchantillonne des couches raster en utilisant l'interpolation
r.resample	Fonctionnalité de reéchantillonage de données raster de GRASS. Vous devez auparavant définir une nouvelle résolution
r.resamp.rst	Reinterpole et calcul l'analyse topographique en utilisant des courbes régularisées avec une tension et un lissage
r.support	Permet la création et/ou la modification des fichiers support de la couche raster
r.support.stats	Met à jour les statistiques du raster
r.proj	Reprojette une couche raster d'une location à la localition actuelle

TAB. 13: Boîte à outils de GRASS : Modules de gestion de la couleur des raster

Modules de gestion de la couleur des raster dans la boîte à outils de GRASS	
Nom du module	**Objectif**
r.colors.table	Définit une table de couleur à partir de tables établies
r.colors.rules	Définit une table de couleur à partir de règles établies
r.colors.rast	Définit une table de couleur d'un raster existant
r.blend	Mélange les composants de couleurs de deux rasters en fonction d'un ratio
r.composite	Mélange le rouge, le vert et le bleu de couches raster pour obtenir un raster d'une couleur
r.his	Génère des couches raster rouge, vert et bleu combinant les valeurs de la teinte, de l'intensité et de la saturation (HIS) à partir de couches raster définit par l'utilisateur en entrée

TAB. 14: Boîte à outils de GRASS : Modules d'analyse spatiale de raster

Modules d'analyse spatiale de raster dans la boîte à outils GRASS	
Nom du module	**Objectif**
r.buffer	Buffer raster
r.mask	Créé un MASK pour limiter les opérations raster
r.mapcalc	Calculateur de couche raster
r.mapcalculator	Algèbre cartographique simple
r.neighbors	Analyse raster des voisins
v.neighbors	Compte les points voisins
r.cross	Crée un produit croisé de la valeur de la catégorie à partir de plus couches raster
r.series	Fait de chaque cellule en sortie une fonction de la valeur attribuée aux cellules correspondantes à la sortie des couches raster
r.patch	Crée une nouvelle couche raster en combinanet d'autres couches raster
r.statistics	Statistiques orienté categories ou objet
r.cost	Renvoie une couche raster montrant le coût cumulatif du déplacement entre des endroits géographiques différents sur une couche raster en entrée dont les valeurs des catégories représentent le coût
r.drain	Trace un flux à travers un modèle d'élévation sur une couche raster
r.shaded.relief	Créé une carte d'ombrage
r.slope.aspect.slope	Génère une carte de pente à partir d'un MNT (Modèle Numérique de Terrain)
r.slope.aspect.aspect	Génère une carte d'aspect à partir d'un MNT (Modèle Numérique de Terrain)
r.param.scale	Extrait les paramètres terrain à partir d'un MNT
r.texture	Génère des images avec des objets de texture à partir d'une couche raster (première série d'indices)
r.texture.bis	Génère des images avec des objets de texture à partir d'une couche raster (seconde série d'indices)
r.los	Analyse raster de la ligne de vue
r.clump	Recatégorise des cellules contig"ue en une catégorie unique
r.grow	Génère une couche raster avec des zones contig"ues augmentées par une cellule
r.thin	Cellules non null minces qui dénote un objet linéaire

TAB. 15: Boîte à outils de GRASS : Modules de gestion des surfaces

Surface management modules in the GRASS Toolbox	
Nom du module	**Objectif**
r.random	Créé une couche vecteur de point aléatoire dans un raster
r.random.cells	Génère des valeurs de cellule aléatoire avec une dépendance spatiale
v.kernel	Densité du noyau Gaussien
r.contour	Produit une couche vectoriel de contours avec des étapes définits à partir d'une couche raster
r.contour2	Produit un contour vectoriel à partir de contours définit par une couche raster
r.surf.fractal	Créé une surface fractal avec une dimension fractale donnée
r.surf.gauss	Module GRASS pour produire une couche raster de déviation gaussienne dont la moyenne et la déviation standard peuvent être exprimé par l'utilisateur
r.surf.random	Produit une couche raster de divation aléatoire uniforme dont le domaine peut être exprimé par l'utilisateur
r.bilinear	Commande d'interpolation bilinéaire pour les couches raster
v.surf.bispline	Interpolation spline bicubique ou bilinéaire avec régularisation de Tykhonov
r.surf.idw	Commande d'interpolation de surface pour des couches raster
r.surf.idw2	Programme de génération de surface
r.surf.contour	Programme de génération de surface à partir de contours rasterisés
v.surf.idw	Interpole les valeurs attributaires (IDW)
v.surf.rst	Interpole les valeurs attributaires (RST)
r.fillnulls	Remplis les zones sans données dans une couche raster en utilisant l'interpolation de splines de v.surf.rst

TAB. 16: Boîte à outils de GRASS : Modules pour changer les valeurs des catégories et des étiquettes des raster

Modules pour changer les valeurs des catégories et des étiquettes des raster dans la boîte à outils de	
Nom du module	**Objectif**
r.reclass.area.greater	Reclasse une couche raster d'une zone supérieure à celle donnée par l'utilisateur (en hectares)
r.reclass.area.lesser	Reclasse une couche raster d'une zone inférieure à celle donnée par l'utilisateur (en hectares)
r.reclass	Reclasse un raster en utilisant un fichier de règles de reclassification
r.recode	Recode des couches raster
r.rescale	Reéchantillonne le domaine des valeurs des catégories d'une couche raster

TAB. 17: Boîte à outils de GRASS : Modules de modélisation hydrologique

Modules de modélisation hydrologique dans la boîte à outils de GRASS	
Nom du module	**Objectif**
r.carve	Utilise des données vecteur de flux, les transforme en raster et extrait la profondeur à partir du MNT en sortie
r.fill.dir	Filtre et génère une couche d'élévation sans dépression et un couche de direction de flux à partir d'une couche d'élévation donnée
r.lake.xy	Remplit le lac à partir de données ponctuelles à un niveau défini
r.lake.seed	Remplit le lac à partir de données à un niveau défini
r.topidx	Crée une carte en 3D basé sur des couches raster et des élévations 2D
r.basins.fill	Génère une couche raster montrant les sous-bassins hydrographiques
r.water.outlet	Programme de création de bassin hydrographique

TAB. 18: Boîte à outils de GRASS : Modules d'analyses statistiques et rapports

Modules d'analyses statistiques et rapports dans la boîte à outils de GRASS	
Nom du module	**Objectif**
r.category	Affiche les valeurs des categories et leus étiquettes avec une couche rster définit par l'utilisateur
r.sum	Réalise la somme des valeurs des cellules d'un raster
r.report	Renvoi des statistiques pour des couches raster
r.average	Trouve la moyenne des valeurs dans une couche de couverture dans des zones assignées de même valeur de catégorie dans une couche définit par l'utilisateur
r.median	Trouve la mediane des valeurs dans une couche de couverture dans des zones assignées de même valeur de catégorie dans une couche définit par l'utilisateur
r.mode	Trouve le mode des valeurs dans une couche de couverture dans des zones assignées de même valeur de catégorie dans une couche définit par l'utilisateur
r.volume	Calcule le volume d'un amas de données, et produit une couche vecteur poncutel GRAS contenant le centro"ide calculé de ces amas
r.surf.area	Estimation de la surface d'une pour des rasters
r.univar	Calcule des statistiques univariées à partir de cellules non nulles d'une couche raster
r.covar	Affiche une matrice de corrélation/covariance pour des couches raster définit par l'utilisateur
r.regression.line	Calcule la régression linéaire à partir de deux cartes raster : y = a + b * x
r.coin	Tabule les occurences mutuelles (co"incidence) des catégories pour deux couches raster

B.5 Modules de données vecteur de la boîte à outils de GRASS

Cette section liste toutes les boîtes de dialogue dans la boîte à outils de GRASS pour utiliser et analyser des données vecteur dans un jeu de données et une région de GRASS sélectionnés.

TAB. 19: Boîte à outils de GRASS : Modules de développement des couches vecteurs

Modules de développement des couches vecteurs de la boîte à outils de GRASS	
Nom du module	**Objectif**
v.build.all	Reconstruit la topôlogie de tous les vecteurs dans le jeu de données
v.clean.break	Coupe les lignes à chaque intersection de la couche vecteur
v.clean.snap	Nettoyage topologique : aimante les lignes vers les sommets en fonction d'un seuil
v.clean.rmdangles	Nettoyage topologique : supprime les noeuds pendants ([NdT] noeuds isolés qui ne ferme pas proprement l'objet)
v.clean.chdangles	Nettoyage topologique : change le type contour d'arc en ligne
v.clean.rmbridge	Supprime les ponts connectant une surface et une ou deux îles
v.clean.chbridge	Change les ponts connectant les surfaces et une ou 2 îles
v.clean.rmdupl	Enlève les lignes dupliquées (faîtes attention aux catégories !)
v.clean.rmdac	Enlève les centro"ides dupliqués des surfaces
v.clean.bpol	Casse les polygones. Les contours sont cassés sur chaque point partagé entre deux ou plus polygones où les angles des segments sont différents
v.clean.prune	Enlève les sommets dans un seuil des lignes et contours
v.clean.rmarea	Enlève les petites surfaces (supprime les contours les plus grand avec des zones adjacantes)
v.clean.rmline	Enlève toutes les lignes ou contours de longueur nulle
v.clean.rmsa	Enlève les petits angles entre les lignes aux niveaux des noeuds
v.type.lb	Convertit des lignes en limites
v.type.bl	Convertit des limites en lignes
v.type.pc	Convertit des points en centroides
v.type.cp	Convertit des centroides en points
v.centroids	Ajoute les centroides manquants aux limites fermées
v.build.polylines	Construit des polylignes à partir de lignes
v.segment	Crée des points/segments à partir de positions et de lignes vectorielles en entrée
v.to.points	Crée des points le longs d'une ligne en entrée
v.parallel	Crée une ligne parallèle à des lignes en entrée
v.dissolve	Dissous des limites dans des zones adjacentes
v.drape	Convertie des vecteurs 2D en vecteur 3D par reéchantillonage de raster d'élélvation
v.transform	Réalise une transformation affine d'une couche vecteur
v.proj	Permet une conversion de la projection de fichier vecteur
v.support	Met à jour les méta-données des couches vecteurs
generalize	Généralisation vectorielle

TAB. 20: Boîte à outils de GRASS : Modules de connexion aux bases de données

Modules de connexion aux bases de données de la boîte à outils de GRASS	
Nom du module	**Objectif**
v.db.connect	Connecte un vecteur à une base de données
v.db.sconnect	Déconnecte un vecteur d'une base de données
v.db.what.connect	Définit/affiche une connexion à une base de données pour un vecteur

TAB. 21: Boîte à outils de GRASS : Modules de modification des champs vectoriels

Modules de modification des champs vectoriels de la boîte à outils de GRASS	
Nom du module	**Objectif**
v.category.add	Ajoute des éléments à la couche (tous les éléments du type de la couche sélectionnée !)
v.category.del	Supprime les valeurs des catégories
v.category.sum	Ajoute une valeur aux valeurs des catégories en cours
v.reclass.file	Reclasse les valeurs des catégories en utilisant un fichier de règles
v.reclass.attr	Reclasse les valeurs des catégories en utilisant une colonne attributaire (entier positif)

TAB. 22: Boîte à outils de GRASS : Travailler avec les modules des vecteurs ponctuels

Travailler avec les modules des vecteurs ponctuels de la boîte à outils de GRASS	
Nom du module	**Objectif**
v.in.region	Crée une nouvelle couche vecteur avec une étendue de la région actuelle
v.mkgrid.region	Crée une grille dans la région actuelle
v.in.db	Importe des points vectoriels d'une table d'une base de données contenant des coordonnées
v.random	Génère aléaoitement une couche de points vectorielle GRASS en 2D/3D
v.kcv	place des points aléatoires dans un jeu test
v.outlier	Supprime les valeurs atypiques des données ponctuelles vectorielles
v.hull	Crée une enveloppe convexe
v.delaunay.area	Triangulation de Delaunay (linéaire)
v.delaunay.area	Triangulation de Delaunay (surface)
v.voronoi.area	Diagramme de Vorono"i (linéaire)
v.voronoi.area	Diagramme de Vorono"i (surface)

TAB. 23: Boîte à outils de GRASS : Modules d'analyse spatiale de vecteur et de réseau

Modules d'analyse spatiale de vecteur et de réseau de la boîte à outils de GRASS	
Nom du module	**Objectif**
v.extract.where	Sélectionne les objets par attributs
v.extract.list	Extrait les objets sélectionnés
v.select.overlap	Sélectione les objets superposés par des objets d'une autre couche
v.buffer	Buffer de vecteur
v.distance	Trouve l'élément le plus proche dans un vecteur 'to' pour des éléments dans un vecteur 'from'
v.net.nodes	Crée des noeuds sur un réseau
v.net.alloc	Alloue un réseau
v.net.iso	Coupe un réseau par des isolignes de coût
v.net.salesman	Connecte des noeuds par la route la plus courte (problème du voyageur de commerce)
v.net.steiner	Connecte des noeuds sélectionnés par l'arbre le plus court (arbre Steiner)
v.patch	Crée une nouvelle couche vecteur par combinaison de couches vecteurs
v.overlay.or	Union de vecteur
v.overlay.and	Intersection de vecteur
v.overlay.not	Soustraction de vecteur
v.overlay.xor	Non intersection de vecteur

TAB. 24: Boîte à outils de GRASS : Mise à jour de vecteur à partir d'autres modules cartographiques

Mise à jour de vecteur à partir d'autres modules cartographiques de la boîte à outils de GRASS	
Nom du module	**Objectif**
v.rast.stats	Calcule des statistiques univariées d'une couche raster GRASS basée sur des objets vecteurs
v.what.vect	Télécharge des cartes pour éditer la table d'attributs
v.what.rast	Télécharge des valeurs raster à la position des points vecteurs vers la table
v.sample	Échantillonne une fichier raster à l'endroit des sites

TAB. 25: Boîte à outils de GRASS : modules de statistique et de rapport de vecteur

Modules de statistique et de rapport de vecteur de la boîte à outils de GRASS	
Nom du module	**Objectif**
v.to.db	Mais des variables de géométrie dans la base de données
v.report	Crée un rapport de statistisque des géométries pour les vecteurs
v.univar	Calcule des statistiques univariées sur la colonne de la table sélectionnée pour une couche vecteur de GRASS
v.normal	Teste la normalité des points

B.6 Modules de données d'imagerie de la boîte à outils de GRASS

Cette section liste toutes les boîtes de dialogue dans la boîte à outils de GRASS pour utiliser et analyser les données d'images dans une région et un jeu de données GRASS sélectionnés.

TAB. 26: Boîte à outils de GRASS : modules analyse d'image

Module d'analyse d'image de la boîte à outils de GRASS	
Nom du module	**Objectif**
i.image.mosaik	Mosaique jusqu'à 4 images
i.rgb.his	Fonction de transformation de la carte de couleur du raster de Rouge Vert Bleu (RVB) en Nuance Intensité Saturation (HIS)
i.his.rgb	Fonction de transformation de la carte de couleur du raster de Nuance Intensité Saturation (HIS) en Rouge Vert Bleu (RVB)
i.landsat.rgb	Balance automatique des couleurs des images LANDSAT
i.fusion.brovey	Transformation de Brovey pour fusionner des cannaux panchromatique multispectrale et de haute résolution
i.zc	Fonction raster de détection de bord vide ([NdT] Zero-crossing edge detection) dans le traitement des images
i.mfilter	
i.tasscap4	Transformation de Tasseled Cap (Kauth Thomas) pour les données LANDSAT-TM 4
i.tasscap4	Transformation de Tasseled Cap (Kauth Thomas) pour les données LANDSAT-TM 5
i.tasscap4	Transformation de Tasseled Cap (Kauth Thomas) pour les données LANDSAT-TM 7
i.fft	Transformation rapide de Fourier (FFT) pour le traitement des images
i.ifft	Transformation inverse rapide de Fourier pour le traitement des images
r.describe	Affiche une liste tierce de valeurs de catégorie trouvé dans une couche raster
r.bitpattern	Compare des motifs d'octets avec une couche raster
r.kappa	Calcule une matrice d'erreur et de paramètre kappa pour l'évaluation de la précision des résultats d'une classification
i.oif	Calcule une table de facteur d'index optimal pour les bandes tm landsat

B.7 Modules de base de données de la boîte à outils de GRASS

Cette section liste toutes les boîtes de dialogue dans la boîte à outils de GRASS pour gérer, se connecter et travailler avec les bases de données externes et internes. Travailler avec des bases de données spatiales externes est possible via OGR n'est pas couvert par ces modules.

TAB. 27: Boîte à outils de GRASS : Modules base de données

Modules de gestion de base de données et d'analyse de la boîte à outils de GRASS	
Nom du module	**Objectif**
db.connect	Définie la connexion à la BdD générale du jeu de données
db.connect.schema	Définie la connexion à la BdD générale avec un schéma du jeu de données
v.db.reconnect.all	Reconnecte un vecteur avec une nouvelle base de données
db.login	Définie un utilisateur/mot de passe pour un pilote/base de données
db.in.ogr	Importe une table d'attribut dans différents formats
v.db.addtable	Créé et ajoute une nouvelle table à un vecteur
v.db.addcol	Ajoute une ou plusieurs colonnes à une table attributaire connectée à une couche vecteur donnée
v.db.dropcol	Supprime une colonne de la table attributaire connectée à une couche vecteur donnée
v.db.renamecol	Renomme une colonne dans une table attributaire connectée à une couche vecteur donnée
v.db.update_const	Permet d'assigner une nouvelle valeur d'une constante à une colonne
v.db.update_query	Permet d'assigner une nouvelle valeur d'une constante à une colonne seulement si le résultat de la requête est TRUE
v.db.update_op	Permet d'assigner une nouvelle valeur, résultat d'une opération sur une ou plusieurs colonne(s), à une colonne dans la table attributaire connectée à une couche donnée
v.db.update_op_query	Permet d'assigner une nouvelle valeur à une colonne, résultat d'une opération sur ou plusieurs colonne(s), seulement si le résutlat de la requête est TRUE
db.execute	Éxecute une requête SQL
db.select	Affiche les résultats d'une sélection d'une base de données basé sur une requête SQL
v.db.select	Affiche les attributs d'une couche vecteur
v.db.select.where	Affiche les attributs d'une couche vecteur avec une requête SQL
v.db.join	Permet de réaliser une jointure de table avec une table d'une couche vecteur
v.db.univar	Calcule des statistiques univariées sur une colonne d'une table sélectionnée pour une couche vecteur de GRASS

B.8 Modules 3D de la boîte à outils de GRASS

Cette section liste toutes les boîtes de dialogues de la boîte à outils de GRASS pour travailler avec les données 3D. GRASS fournit plus de modules, mais ils sont actuellement seulement disponibles en utilisant la console de GRASS.

TAB. 28: Boîte à outils de GRASS : visualisation 3D

Modules de visualisation 3D et d'analyses de la boîte à outils de GRASS	
Nom du module	**Objectif**
nviz	Vue 3D dans nviz

B.9 Modules d'aide de la boîte à outils de GRASS

Le manuel de référence du SIG GRASS offre un aperçu complet des modules de GRASS disponibles, non limité aux modules et leurs fonctionnalités souvent limités implémenté dans la boîte à outils de GRASS.

TAB. 29: Boîte à outils de GRASS : manuel de référence

Modules de manuel de référence de la boîte à outils de GRASS	
Nom du module	**Objectif**
g.manual	Affiche la page HTML du manuel de GRASS

C Guide d'installation

Ce chapitre comporte les informations nécessaires à la compilation et à l'installation de QGIS 1.0.0. Ce document est une conversion LaTeX du fichier INSTALL.t2t embarqué dans les sources de QGIS depuis le 16 décembre 2008.

Une version est maintenue continuellement sur le wiki : http ://wiki.qgis.org/qgiswiki/BuildingFromSource

(http ://www.cmake.org)

C.1 General Build Notes

Depuis la version 0.8.1, QGIS n'utilise plus les autotools pour la compilation mais le logiciel cmake (http ://www.cmake.org). Le script de configuration vérifie seulement la présence de cmake.

Pour plus d'informations :

http ://wiki.qgis.org/qgiswiki/Building_with_CMake

C.2 Un aperçu des dépendances requises

Dépendances requises pour la compilation :

- CMake >= 2.4.3
- Flex, Bison

Dépendances requises pour l'exécution :

- Qt >= 4.3.0
- Proj >= ? (fonctionne avec la 4.4.x)
- GEOS >= 2.2 (3.0 est supporté, la 2.1.x marche peut-etre)
- Sqlite3 >= ? (probablement 3.0.0)
- GDAL/OGR >= 1.4.x

Dépendances optionnelles :

- pour l'extension GRASS plugin - GRASS >= 6.0.0
- pour le géoreferencement - GSL >= ? (marche avec la 1.8)
- pour le support de PostGIS et de l'extension SPIT - PostgreSQL >= 8.0.x
- pour l'extension GPS - expat >= ? (1.95 is OK)
- pour l'exportation vers mapserver PyQGIS - Python >= 2.3 (2.5+ preferred)
- pour PyQGIS - SIP >= 4.5, PyQt >= doit s'accorder avec la version de Qt

Dépendances recommandées :

– pour l'extension GPS - gpsbabel

D Compiler sous Windows avec msys

Note : Pour connaître dans le détail la manière de compiler vous-même les dépendances, vous pouvez visiter le site de Marco Pasetti :

http ://www.webalice.it/marco.pasetti/qgis+grass/BuildFromSource.html

D.1 MSYS :

MSYS fournit un environment de compilation similaire à celui d'Unix mais sous Windows. Nous avons créé une archive zip contenant toutes les dépendances, à télécharger ici :

http ://download.osgeo.org/qgis/win32/msys.zip

et à extraire vers c :\msys

Si vous voulez configurer votre environnement msys vous-même, de plus amples instructions sont disponibles dans ce document.

D.2 Qt4.3

Téléchargez l'édition open source de Qt 4.3 (incluant le téléchargement et l'installation de MinGW) à cette adresse :

http ://www.trolltech.com/developer/downloads/qt/windows

Quand l'installateur vous demande MinGW, vous n'avez pas besoin de le télécharger et de l'installer, mais juste de pointer vers le répertoire c :\msys\mingw

Quand l'installation de Qt est terminée :

Éditez C :\Qt\4.3.0\bin\qtvars.bat et rajoutez les lignes suivantes :

```
set PATH=%PATH%;C:\msys\local\bin;c:\msys\local\lib
set PATH=%PATH%;"C:\Program Files\Subversion\bin"
```

Je vous sugère également d'ajouter C :\Qt\4.3.0\bin\ aux variables d'environnements de votre système.

Si vous comptez faire du débuggage, vous devrez compiler une version debug de Qt : C :\Qt\4.3.0\bin\qtvars.bat compile_debug

Note : il existe un problème lors de la compilation d'une version debug de Qt 4.3, le script se finit avec le message 'minggw32-make : *** No rule to make target 'debug'. Stop.". Pour réussir la compilation, vous devez allez dans le répertoire src et lancez la commande suivante :

```
c:\Qt\4.3.0 make
```

D.3 Python : (optionnel)

Suivez cette section si vous souhaitez utiliser Python pour QGIS. Afin de pouvoir utiliser Python pour QGIS, vous avez besoin de compiler SIP et PyQt4 à partir des sources car leurs programmes d'installation n'incluent pas certains fichiers qui sont nécessaires.

D.3.1 Télécharger et installer Python - utilisation de l'installateur sous Windows

(le répertoire d'installation n'a aucune importance)

http ://python.org/download/

D.3.2 Télécharger SIP et les sources de PyQt4

```
http://www.riverbankcomputing.com/software/sip/download
http://www.riverbankcomputing.com/software/pyqt/download
```

Il faut extraire chacune des archives zip dans un répertoire temporaire. Assurez-vous d'avoir des versions qui correspondent bien à votre version de Qt.

D.3.3 Compiler SIP

```
c:\Qt\4.3.0\bin\qtvars.bat
python configure.py -p win32-g++
make
make install
```

D.3.4 Compiler PyQt

```
c:\Qt\4.3.0\bin\qtvars.bat
python configure.py
make
make install
```

D.3.5 Notes finales pour Python

Note : Vous pouvez effacer les répertoires contenants les fichiers désarchivés de SIP et PyQt4 après l'installation, ils ne sont plus nécessaires.

D.4 Subversion :

De manière à obtenir les sources depuis le répertoire svn de QGIS, vous avez besoin de Subversion :

http ://subversion.tigris.org/files/documents/15/36797/svn-1.4.3-setup.exe

D.5 CMake :

CMake est le système de compilation de Quantum GIS. Téléchargez le ici :

http ://www.cmake.org/files/v2.4/cmake-2.4.6-win32-x86.exe

D.6 QGIS :

Lancez une fenêtre cmd.exe (Démarrer -> Exécuter -> cmd.exe). Créez un répertoire de développement et déplacez dans ce répertoire :

```
md c:\dev\cpp
cd c:\dev\cpp
```

Obtenez les dernières versions des sources avec svn :

```
svn co https://svn.osgeo.org/qgis/trunk/qgis
```

Pour la branche 1.0

```
svn co https://svn.osgeo.org/qgis/branches/Version-1_0
```

D.7 Compiler :

Pour votre culture, vous pouvez lire à la fin de ce document quelques lignes sur la façon de compiler avec CMake.

Lancez une fenêtre cmd.exe (Démarrer -> Exécuter -> cmd.exe) si vous ne l'avez pas encore fait. Ajoutez les chemins vers le compilateur et l'environnement MSYS :

```
c:\Qt\4.3.0\bin\qtvars.bat
```

Pour faciliter les choses, ajoutez aussi c :\Qt\4.3.0\bin\ pour que vous n'ayez qu'à taper qtvars.bat dans la console. Créez le répertoire de compilation et établissez-le en tant que répertoire courant :

```
cd c:\dev\cpp\qgis
md build
cd build
```

D.8 Configuration

```
cmakesetup ..
```

Note : Vous devez inclure les '..' marqués ci-dessus.

Cliquez sur le bouton 'Configure'. lorsque c'est demandé, vous devez choisir 'MinGW Makefiles' comme générateur.

Sous Windows 2000, du fait d'un bug avec MinGW choisissez plutôt 'MSYS Makefiles'.

Toutes les dépendances devraient être sélectionnées automatiquement si vous avez spécifié les chemins correctement. La seule chose que vous ayez à changer est la destination de l'installation avec (CMAKE_INSTALL_PREFIX)et/ou 'Debug'.

Pour plus de compatibilité avec les scripts NSIS, je vous recommande de laisser le préfixe d'installation par défaut c :\program files\

Quand la configuration est terminée, cliquez sur OK pour quitter l'utilitaire d'installation.

D.9 Compilation et installation

```
 make make install
```

D.10 Lancez qgis.exe depuis son répertoire d'installation (CMAKE_INSTALL_PREFIX)

Copiez toutes les .dll :s nécessaire dans le même répertoire que l'exécutable de qgis ou QGIS signalera des bibliothèques absentes lors de son lancement.

La meilleure façon de procéder est de télécharger l'installateur de QGIS depuis http ://qgis.org/uploadfiles/testbuilds/ et de l'installer. Maintenant, copiez le répertoire d'installation C :\Program Files\Quantum GIS vers c :\Program Files\qgis-0.8.1 (ou tout autre numéro de version. Le nom doit être strictement identique au numéro de version.). Une fois que cela est fait, vous pouvez désinstaller la version téléchargée. Vérifiez que le répertoire correspondant n'apparaît plus dans c :\Program Files

Une autre possibilité est de lancer qgis.exe quand vos chemins comportent les répertoires c :\msys\local\bin et c :\msys\local\lib, auquel cas les DLLs seront utilisées depuis ces emplacements.

D.11 Création du fichier d'installation : (optionnel)

Téléchargez et installez NSIS depuis (http ://nsis.sourceforge.net/Main_Page)

En passant par l'explorateur de fichier, entrez dans le répertoire win_build se situant dans l'arborescence des sources de QGIS. Lisez le fichier README et suivez les instructions. Ensuite, faites un clic droit sur qgis.nsi et sélectionnez l'option 'Compile NSIS Script'.

E Compilation sous Mac OSX en utilisant XCODE et cmake (QGIS > 0.8)

Avec cette approche, nous allons essayer d'éviter de construire les dépendances depuis les sources, mais plutôt utiliser ce qui est déjà disponible.

Quelques notes pour compiler sous Mac OS X 10.5 (Leopard) sont également disponibles.

E.1 Installer XCODE

Je vous recommande d'obtenir la dernière version de xcode depuis le site Apple XDC.

Note : Il sera peut être nécessaire de créer des liens symboliques après l'installation (surtout si vous utilisez xcode 2.5 sur Tiger) :

```
cd /Developer/SDKs/MacOSX10.4u.sdk/usr/
sudo mv local/ local_
sudo ln -s /usr/local/ local
```

E.2 Installer Qt4 depuis un .dmg

Vous aurez besoin d'une version au moins égale à Qt4.3.0 bien que je vous suggère de prendre la dernière version en date.

```
ftp://ftp.trolltech.com/qt/source/qt-mac-opensource-4.3.2.dmg
```

Si vous voulez les bibliothèques de débogage, Qt fournit aussi un .dmg les contenant :

```
ftp://ftp.trolltech.com/qt/source/qt-mac-opensource-4.3.2-debug-libs.dmg
```

Une fois le téléchargement effectué, lancez l'installateur.

Note : vous avez besoin des accès administrateurs pour l'installation.

Après installation, vous devez faire deux modifications minimes :

Ouvrez `/Library/Frameworks/QtCore.framework/Headers/qconfig.h` et changez

Note : cela ne semble plus nécessaire depuis la version 4.2.3

`QT_EDITION_Unknown` pour `QT_EDITION_OPENSOURCE`

Le second changement porte sur le lien symbolique de mkspec pour qu'il pointe sur macx-g++ :

```
cd /usr/local/Qt4.3/mkspecs/
sudo rm default
sudo ln -sf macx-g++ default
```

E.3 Installer l'environnement de développement pour les dépendances de QGIS

Téléchargez et installez l'excellent environnement de William Kyngesburye qui inclut proj, gdal, sqlite3 etc

```
http://www.kyngchaos.com/wiki/software:frameworks
```

William fournit un autre installateur pour Postgresql/PostGIS. Il est disponible ici :

http://www.kyngchaos.com/wiki/software:postgres

Il y a d’autres dépendances qui ne sont pas incluses (au moment où le manuel a été écrit) et que nous devrons compiler depuis les sources.

E.3.1 Dépendances supplémentaires : GSL

Téléchargez la Bibliothèque Scientifique Gnu (Gnu Scientific Library)

```
curl -O ftp://ftp.gnu.org/gnu/gsl/gsl-1.8.tar.gz
```

Puis extrayez là et faites la compilation vers un préfixe de /usr/local :

```
tar xvfz gsl-1.8.tar.gz
cd gsl-1.8
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

E.3.2 Dépendances supplémentaires : Expat

Obtenez les sources d’expat :

http://sourceforge.net/project/showfiles.php?group_id=10127

```
tar xvfz expat-2.0.0.tar.gz
cd expat-2.0.0
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

E.3.3 Dépendances supplémentaires : SIP

Soyez sûr d’avoir la dernière version de Python

http://www.python.org/download/mac/

Note : Leopard inclut déjà Python 2.5. Vous pouvez cependant utiliser la version de python.org .

Téléchargez les liens python de SIP à

http://www.riverbankcomputing.com/software/sip/download

Puis extrayez-les et faites la compilation (l'installation se fera par défaut dans l'environnement Python) :

```
tar xvfz sip-<version number>.tar.gz
cd sip-<version number>
python configure.py
make
sudo make install
cd ..
```

Notes

Si vous compilez sous Leopard avec la version de Python embarquée, SIP voudra s'installer dans le chemin système – ce qui n'est pas une bonne idée. Utilisez cette commande de configuration à la place la commande précédente :

```
python configure.py -d /Library/Python/2.5/site-packages -b \
/usr/local/bin -e /usr/local/include -v /usr/local/share/sip
```

E.3.4 Dépendances supplémentaires : PyQt

Si vous rencontrez des problèmes pour compiler PyQt en suivant les instructions suivantes, vous pouvez essayez d'ajouter explicitement le chemin vers votre environnement python :

```
export PATH=/Library/Frameworks/Python.framework/Versions/Current/bin:$PATH$
```

Obtenez les liens python pour Qt depuis

http://www.riverbankcomputing.com/software/pyqt/download

Puis extrayez et compilez :

```
tar xvfz PyQt-mac<version number here>
cd PyQt-mac<version number here>
export QTDIR=/Developer/Applications/Qt
python configure.py
yes
make
sudo make install
cd ..
```

Notes

Si vous compilez sous Leopard avec la version de Python embarquée, PyQt voudra s'installer dans le chemin système – ce qui n'est pas une bonne idée. Utilisez cette commande de configuration à la place la commande précédente :

```
python configure.py -d /Library/Python/2.5/site-packages -b /usr/local/bin
```

Un problème avec des symboles non définis dans QtOpenGL peut se présenter sous Leopard. Éditez QtOpenGL/makefile et ajoutez -undefined dynamic_lookup aux LFLAGS.

E.3.5 Dépendances supplémentaires : Bison

Note : Leopard inclut Bison 2.3, cette étape peut être ignorée.

La version par défaut de bison sous Mac OSX est trop ancienne ancienne pour vos besoins, télécharger une version plus récente :

```
curl -O http://ftp.gnu.org/gnu/bison/bison-2.3.tar.gz
```

Maintenant compilez et installez dans un préfix de /usr/local :

```
tar xvfz bison-2.3.tar.gz
cd bison-2.3
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

E.4 Installer CMAKE pour OSX

Obtenez la dernière version à :

```
http://www.cmake.org/HTML/Download.html
```

La version utilisée au moment de l'écriture de cette partie est :

```
curl -O http://www.cmake.org/files/v2.4/cmake-2.4.6-Darwin-universal.dmg
```

Une fois téléchargé, installez le fichier .dmg

E.5 Installer subversion pour OSX

Note : Leopard inclut SVN, cette étape peut donc être ignorée sous Leopard.

Ce http://sourceforge.net/projects/macsvn/ projet a une version téléchargeable de svn. Si vous préférez une interface, pensez à prendre leur client graphique. Téléchargez l'interface console ici :

```
curl -O http://ufpr.dl.sourceforge.net/sourceforge/macsvn/Subversion_1.4.2.zip
```

Vous devez également installer BerkleyDB disponible à http://sourceforge.net/projects/macsvn/.

```
curl -O
http://ufpr.dl.sourceforge.net/sourceforge/macsvn/Berkeley_DB_4.5.20.zip
```

Une fois encore, extrayez l'archive zip et exécuter l'installateur qu'elle contient. Enfin, nous devons nous assurer que la commande svn a le bon chemin. Ajoutez la ligne suivante à la fin de /etc/bashrc en utilisant sudo :

```
sudo vim /etc/bashrc
```

Et ajoutez cette ligne en bas avant de sauvegarder et quitter :

```
export PATH=/usr/local/bin:$PATH:/usr/local/pgsql/bin
```

/usr/local/bin doit être le premier sur le chemin pour que la nouvelle version de bison (que nous allons compiler par la suite) soit trouvée avant celle embarquée par MacOSX.

Maintenant, fermez et rouvrez votre console pour bénéficier des variables à jour.

E.6 Obtenir QGIS avec SVN

Nous allons maintenant vérifier les sources de QGIS. Nous allons d'abord créer un répertoire de travail :

```
mkdir -p ~/dev/cpp cd ~/dev/cpp
```

Maintenant nous téléchargeons les sources :

Pour la branche principale :

```
svn co https://svn.osgeo.org/qgis/trunk/qgis qgis
```

Pour la branche de la version

```
svn co https://svn.osgeo.org/qgis/branches/Version-1_0
```

Au premier téléchargement, vous allez sans doute avoir le message suivant :

```
Error validating server certificate for 'https://svn.qgis.org:443':
- The certificate is not issued by a trusted authority. Use the fingerprint to
  validate the certificate manually!  Certificate information:
- Hostname: svn.qgis.org
- Valid: from Apr  1 00:30:47 2006 GMT until Mar 21 00:30:47 2008 GMT
- Issuer: Developer Team, Quantum GIS, Anchorage, Alaska, US
- Fingerprint: 2f:cd:f1:5a:c7:64:da:2b:d1:34:a5:20:c6:15:67:28:33:ea:7a:9b
  (R)eject, accept (t)emporarily or accept (p)ermanently?
```

Je vous recommande d'appuyer sur 'p' pour accepter la clé de manière permanente.

E.7 Configurer la compilation

Nous allons créer un dossier 'build' pour le processus de compilation. Par convention, je compile mes logiciels dans un répertoire 'apps' présent dans mon dossier personnel /home. Si les permissions sont accrodées, vous pouvez directement compiler dans votre répertoire /Applications. Les instructions qui suivent présupposent que vous travaillez dans le dossier ${HOME}/apps...

```
cd qgis
```

```
mkdir build
cd build
cmake -D CMAKE_INSTALL_PREFIX=$HOME/apps/ -D CMAKE_BUILD_TYPE=Release ..
```

Note : Pour trouver l'installation personnalisée de SIP sous Leopard, ajoutez ""-D SIP_BINARY_-PATH=/usr/local/bin/sip"" à la commande cmake ci-dessus, avant les .. de la fin :

```
cmake -D CMAKE_INSTALL_PREFIX=$HOME/apps/ -D CMAKE_BUILD_TYPE=Release -
D SIP_BINARY_PATH=/usr/local/bin/sip ..
```

Pour utiliser GRASS depuis le dossier Applications sous OSX, vous pouvez utilisez l'option cmake suivante (GRASS 6.3 est la version minimum requise, modifier le numéro au besoin) :

```
cmake -D CMAKE_INSTALL_PREFIX=${HOME}/apps/ \
      -D GRASS_INCLUDE_DIR=/Applications/GRASS-6.3.app/Contents/MacOS/
      include \
      -D GRASS_PREFIX=/Applications/GRASS-6.3.app/Contents/MacOS \
      -D CMAKE_BUILD_TYPE=Release \
      ..
```

Ou, dans un style Unix, utilisez l'option suivante :

```
cmake -D CMAKE_INSTALL_PREFIX=${HOME}/apps/ \
  -D GRASS_INCLUDE_DIR=/user/local/grass-6.3.0/include \
  -D GRASS_PREFIX=/user/local/grass-6.3.0 \
  -D CMAKE_BUILD_TYPE=Release \
  ..
```

E.8 Compilation

maintenant nous pouvons démarrer la compilation de QGIS :

```
make
```

Si tout se passe sans erreurs ou avertissements, vous pouvez l'installer avec :

```
make install
```

F Compilation sous GNU/Linux

F.1 Compiler QGIS avec Qt4.x

Requiert : Ubuntu Hardy / Debian

Ces notes sont pour Ubuntu 7.10 - d'autres versions ou distributions dérivées de Debian peuvent nécessiter de légères adaptations.

Le but principal est de vous montrer que l'on compiler QGIS en utilisant exclusivement des paquets binaires pour ***toutes*** les dépendances - en n'utilisant les sources que pour le coeur de QGIS. Je préfère cette approche, car elle signifie que nous pouvons concentrer sur QGIS en laissant apt se dépatouiller pour toute la gestion.

Ce document part du principe que votre installation vient juste d'être installée. Si vous l'avez déjà utilisée pendant quelque temps, vous aurez sans doute à sauter quelques passages inutiles.

F.2 Préparer apt

Les paquets desquels dépend QGIS sont disponibles dans la source "universe"d'Ubuntu. Vous devez l'activer manuellement :

1. Éditez votre fichier /etc/apt/sources.list 2. Décommentez toutes les lignes début par "deb"

Vous aurez besoin d'utiliser (K)Ubuntu 'edgy' ou supérieur pour que toutes les dépendances puissent être résolues.

Mettez à jour les bases de données des sources :

```
sudo apt-get update
```

F.3 Installer Qt4

```
sudo apt-get install libqt4-core libqt4-debug  \
libqt4-dev libqt4-gui libqt4-qt3support libqt4-sql lsb-qt4 qt4-designer \
qt4-dev-tools qt4-doc qt4-qtconfig uim-qt gcc libapt-pkg-perl resolvconf
```

Note spéciale : Si vous suivez ce guide sur un système où les outils de développements de Qt3 sont déjà installés, vous risquez des conflits avec Qt4 notemment avec qmake. Par exemple, qmake peut pointer vers Qt3 et par vers Qt4. Les pacquets Ubuntu Qt3 et Qt4 sont pour être indépedants. Ubuntu gère cette situation en fournissant 3 exécutables différents pour qmake :

```
/usr/bin/qmake -> /etc/alternatives/qmake
/usr/bin/qmake-qt3
/usr/bin/qmake-qt4
```

La même règle s'applique à tous les autres exécutables de Qt. Vous remarquerez que le nom générique 'qmake' est géré par apt. Avant de compiler QGIS, il faut s'assurer les outils Qt4 soient sélectionnés. Pour revenir à la version antérieure Qt3, vous pouvez utiliser le même processus.

```
sudo update-alternatives --config qmake
sudo update-alternatives --config uic
sudo update-alternatives --config designer
sudo update-alternatives --config assistant
sudo update-alternatives --config qtconfig
sudo update-alternatives --config moc
sudo update-alternatives --config lupdate
sudo update-alternatives --config lrelease
sudo update-alternatives --config linguist
```

Utilisez le dialogue console qui se présente après chacune de ces commandes pour définir la version de Qt pour chacune des applications.

F.4 Installer les dépendances additionnelles requises par QGIS

```
sudo apt-get install gdal-bin libgdal1-dev libgeos-dev proj \
libgdal-doc libhdf4g-dev libhdf4g-run python-dev \
libgsl0-dev g++ libjasper-dev libtiff4-dev subversion \
libsqlite3-dev sqlite3 ccache make libpq-dev flex bison cmake txt2tags \
python-qt4 python-qt4-dev python-sip4 sip4 python-sip4-dev
```

Note : Les utilisateurs Debian devraient plutôt utiliser libgdal-dev ci-dessus.

Note : Pour les liens python, SIP $>=$ 4.5 and PyQt4 $>=$ 4.1 sont requis. Certaines distributions (e.g. Debian ou SuSE) fournissent seulement SIP $<$ 4.5 and PyQt4 $<$ 4.1. Il vous faudra compiler ces paquets depuis les sources.

Si cmake n'est pas encore installé :

```
sudo apt-get install cmake
```

F.5 Etapes spécifiques à GRASS

Note : Si vous n'avez pas besoin du support de GRASS, vous pouvez passer cette partie.

Vous pouvez installer GRASS depuis dapper :

```
sudo apt-get install grass libgrass-dev libgdal1-1.4.0-grass
```

/ !\ Vous aurez peut-être à définir votre version de grass, e.g. libgdal1-1.3.2-grass

F.6 Installer ccache (Optionnel)

Vous pouvez installer ccache pour accélérer la compilation :

```
cd /usr/local/bin
sudo ln -s /usr/bin/ccache gcc
sudo ln -s /usr/bin/ccache g++
```

F.7 Preparer votre environnement de développement

Par convention, je fais mon développement dans $HOME/dev/<langage>, donc dans le cas présent nous allons créer un répertoire pour le C++ : Ce répertoire sera valable pour toutes les commandes qui suivent.

```
mkdir -p ${HOME}/dev/cpp
cd ${HOME}/dev/cpp
```

F.8 Obtenir le code source de QGIS

Il ya deux façons de faire, utilisez la méthode anonyme si vous n'avez aucun droit d'édition sur le répertoire svn de QGIS, ou la méthode développeur si vous pouvez soumettre des changements aux sources.

1. Méthode anonyme

```
cd ${HOME}/dev/cpp
svn co https://svn.osgeo.org/qgis/trunk/qgis qgis
```

2.Méthode développeur

```
cd ${HOME}/dev/cpp
svn co --username <yourusername> https://svn.osgeo.org/qgis/trunk/qgis qgis
```

Au premier essai, il vous faudra accepter le certificat de sécurité de QGIS. Appuyez sur 'p' pour que ce soit permanent :

```
Error validating server certificate for 'https://svn.qgis.org:443':
   - The certificate is not issued by a trusted authority. Use the
     fingerprint to validate the certificate manually!  Certificate
     information:
   - Hostname: svn.qgis.org
   - Valid: from Apr  1 00:30:47 2006 GMT until Mar 21 00:30:47 2008 GMT
   - Issuer: Developer Team, Quantum GIS, Anchorage, Alaska, US
   - Fingerprint:
     2f:cd:f1:5a:c7:64:da:2b:d1:34:a5:20:c6:15:67:28:33:ea:7a:9b (R)eject,
     accept (t)emporarily or accept (p)ermanently?
```

F.9 Commencer la compilation

Note : La section suivante explique comment faire un paquet Debian

Je compile les versions de développements de QGIS dans mon répertoire ˜/apps pour éviter des conflits avec les paquets Ubuntu présents dans /usr. De cette manière, vous pouvez utiliser une version de développement en même temps que la version la distribution. Je vous conseille de faire de même :

```
mkdir -p ${HOME}/apps
```

maintenant créons un répertoire de travail et lançons ccmake :

```
cd qgis
mkdir build
cd build
ccmake ..
```

Quand nous lançons ccmake (les .. sont requis), un menu apparaît pour vous laisser configurer différents aspects de la compilation. Si vous n'avez pas l'accès root ou ne voulez pas écraser une version

existante, définissez CMAKE_BUILD_PREFIX vers un autre emplacement (/home/timlinux/apps). Maintenant appuyez sur ’c’ pour configurer, ’e’ pour écarter les différents messages d’erreurs et ’g’ pour générer les fichiers make. **Note :** quelquefois il faut appuyez sur ’c’ à plusieurs reprises pour permettre de disposer de ’g’. Quand la génération est terminée, appuyez sur ’q’ pour sortir du dialogue interactif de ccmake.

passons à la compilation :

```
make
make install
```

Cela peut prendre plus ou moins de temps selon votre ordinateur.

F.10 Construire un paquet Debian

Au lieu de créer une installation personnalisée comme dans les étapes précédentes, vous créer un paquet Debian réutilisable. Cela se fait à partir du répertoire racine QGIS, où vous trouverez un répertoire debian.

vous avez d’abord besoin d’installer les outils de paquetage de Debian :

```
apt-get install build-essential
```

Le paquet QGIS se crée avec la commande suivante :

```
dpkg-buildpackage -us -us -b
```

Note : Si `dpkg-buildpackage` se plaint de dépendances non résolues, vous devrez les installer avec `apt-get` et relancer la commande.

Note : Si vous avez installé `libqgis1-dev`, vous devez l’effacer avec `dpkg -r libqgis1-dev` ou `dpkg-buildpackage` se plaindra d’un conflit.

Les paquets sont créés dans le répertoire Parent (ie, un niveau au-dessus). Installez-les avec dpkg :

```
sudo dpkg -i \
 ../qgis_1.0preview16_amd64.deb \
 ../libqgis-gui1_1.0preview16_amd64.deb \
 ../libqgis-core1_1.0preview16_amd64.deb \
 ../qgis-plugin-grass_1.0preview16_amd64.deb \
 ../python-qgis_1.0preview16_amd64.deb
```

F.11 Lancer QGIS

Vous pouvez essayer d'exécuter QGIS :

```
$HOME/apps/bin/qgis
```

Si tout à bien fonctionné, l'application devrait apparaître sur votre écran.

G Création d'un environnement MSYS pour la compilation de Quantum GIS

G.1 Installation initiale

G.1.1 MSYS

Cet environnement fournit des utilitaires UNIX pour Windows et est nécessaire pour compiler nombre de dépendances.

`http://puzzle.dl.sourceforge.net/sourceforge/mingw/MSYS-1.0.11-2004.04.30-1.exe`

Installez le dans `c :\msys`

Tout ce que nous allons compiler sera placé dans ce répertoire.

G.1.2 MinGW

A télécharger depuis :

`http://puzzle.dl.sourceforge.net/sourceforge/mingw/MinGW-5.1.3.exe`

Installez le à `c :\msys\mingw`

Il suffit de télécharger et d'installer les composants `g++` et `mingw-make`.

G.1.3 Flex et Bison

Flex et Bison sont des outils indispensables à la compilation de GRASS et QGIS.

Téléchargez les paquets suivants :

http://gnuwin32.sourceforge.net/downlinks/flex-bin-zip.php

http://gnuwin32.sourceforge.net/downlinks/bison-bin-zip.php

http://gnuwin32.sourceforge.net/downlinks/bison-dep-zip.php

extrayez les vers `c :\msys\local`

G.2 Installation des dépendances

G.2.1 Préparation

Paul Kelly a fourni un énorme travail en préparant un paquet contenant des bibliothèques précompilées pour GRASS :

- zlib-1.2.3
- libpng-1.2.16-noconfig
- xdr-4.0-mingw2
- freetype-2.3.4
- fftw-2.1.5
- PDCurses-3.1
- proj-4.5.0
- gdal-1.4.1

C'est disponible à :

http://www.stjohnspoint.co.uk/grass/wingrass-extralibs.tar.gz

Et pour ceux qui sont intéressés, il a laissé des notes sur la façon de les compiler :

http://www.stjohnspoint.co.uk/grass/README.extralibs

extrayez le tout vers `c :\msys\local`

G.2.2 GDAL étape une

Puisque Quantum GIS nécessite GDAL avec un support GRASS, nous avons besoin de compiler GDAL depuis les sources, car le paquet de Paul Kelly ne contient pas ce support. Le programme est le suivant :

1. compiler GDAL sansGRASS
2. compiler GRASS
3. compiler GDAL avec GRASS

Donc, commençons par télécharger les sources de GDAL :

http://download.osgeo.org/gdal/gdal141.zip

Extrayez le dans un répertoire, de préférence c :\msys\local\src.

Lancez une console MSYS, allez dans le répertoire de gdal-1.4.1 et exécutez la commande suivante. Vous pouvez toutes les placer dans un script shell, p. ex. build-gdal.sh, puis le lancer. Cette procédure permet de s'assurer que la bibliothèque est bien créée en tant que DLL et que les programmes lui seront dynamiquement liés.

```
CFLAGS="-O2 -s" CXXFLAGS="-O2 -s" LDFLAGS=-s ./configure --without-libtool \
--prefix=/usr/local --enable-shared --disable-static --with-libz=/usr/local \
--with-png=/usr/local
make
make install
rm /usr/local/lib/libgdal.a
g++ -s -shared -o ./libgdal.dll -L/usr/local/lib -lz -lpng ./frmts/o/*.o
./gcore/*.o \
./port/*.o ./alg/*.o ./ogr/ogrsf_frmts/o/*.o ./ogr/ogrgeometryfactory.o \
./ogr/ogrpoint.o ./ogr/ogrcurve.o ./ogr/ogrlinestring.o ./ogr/ogrlinearring.o \
./ogr/ogrpolygon.o ./ogr/ogrutils.o ./ogr/ogrgeometry.o
./ogr/ogrgeometrycollection.o \
./ogr/ogrmultipolygon.o ./ogr/ogrsurface.o ./ogr/ogrmultipoint.o \
./ogr/ogrmultilinestring.o ./ogr/ogr_api.o ./ogr/ogrfeature.o
./ogr/ogrfeaturedefn.o \
./ogr/ogrfeaturequery.o ./ogr/ogrfeaturestyle.o ./ogr/ogrfielddefn.o \
./ogr/ogrspatialreference.o ./ogr/ogr_srsnode.o ./ogr/ogr_srs_proj4.o \
./ogr/ogr_fromepsg.o ./ogr/ogrct.o ./ogr/ogr_opt.o ./ogr/ogr_srs_esri.o \
./ogr/ogr_srs_pci.o ./ogr/ogr_srs_usgs.o ./ogr/ogr_srs_dict.o
./ogr/ogr_srs_panorama.o \
./ogr/swq.o ./ogr/ogr_srs_validate.o ./ogr/ogr_srs_xml.o
./ogr/ograssemblepolygon.o \
./ogr/ogr2gmlgeometry.o ./ogr/gml2ogrgeometry.o
install libgdal.dll /usr/local/lib
cd ogr
g++ -s ogrinfo.o -o ogrinfo.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s ogr2ogr.o -o ogr2ogr.exe -lgdal -L/usr/local/lib -lpng -lz -lgdal
g++ -s ogrtindex.o -o ogrtindex.exe -lgdal -L/usr/local/lib -lpng -lz -lgdal
install ogrinfo.exe ogr2ogr.exe ogrtindex.exe /usr/local/bin
cd ../apps
g++ -s gdalinfo.o -o gdalinfo.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdal_translate.o -o gdal_translate.exe -L/usr/local/lib -lpng -lz -lgdal
```

```
g++ -s gdaladdo.o -o gdaladdo.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdalwarp.o -o gdalwarp.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdal_contour.o -o gdal_contour.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdaltindex.o -o gdaltindex.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdal_rasterize.o -o gdal_rasterize.exe -L/usr/local/lib -lpng -lz -lgdal
install gdalinfo.exe gdal_translate.exe gdaladdo.exe gdalwarp.exe
gdal_contour.exe \
gdaltindex.exe gdal_rasterize.exe /usr/local/bin
```

Finalement, éditez manuellement `gdal-config` dans `c :\msys\local\bin` pour remplacer les références statiques par -lgdal :

```
CONFIG_LIBS="-L/usr/local/lib -lpng -lz -lgdal"
```

la procédure de compilation de GDAL peut être simplifié en utilisant libtool grâçe à un petit rectificatif, configurer gdal comme il suit : ./configure –with-ngpython –with-xerces=/local/ –with-jasper=/local/ –with-grass=/local/grass-6.3.cvs/ –with-pg=/local/pgsql/bin/pg_config.exe

puis corriger libtool avec : mv libtool libtool.orig cat libtool.orig | sed 's/max_cmd_len=8192/max_-cmd_len=32768/g' > libtool

Libtool sous Windows gére une longueur de ligne de 8192 et essaye de rechercher les liaisons mais il se plante lamentablement. C'est le moyen de le contourner.

Make et make install devraient fonctionner sans souci après ça.

G.2.3 GRASS

Obtenez les sources depuis CVS ou téléchargez un instantané hebdomadaire :

> http ://grass.itc.it/devel/cvs.php

Dans la console MSYS allez dans le répertoire où vous avez placé les sources (e.g. `c :\msys\local\src\grass-6.3.cvs`)

Lancez ces commandes :

```
export PATH="/usr/local/bin:/usr/local/lib:$PATH"
./configure --prefix=/usr/local --bindir=/usr/local
--with-includes=/usr/local/include \
```

```
--with-libs=/usr/local/lib --with-cxx --without-jpeg --without-tiff
--with-postgres=yes \
--with-postgres-includes=/local/pgsql/include --with-pgsql-libs=/local/pgsql/lib
\
--with-opengl=windows --with-fftw --with-freetype \
--with-freetype-includes=/mingw/include/freetype2 \
--without-x --without-tcltk \
--enable-x11=no --enable-shared=yes --with-proj-share=/usr/local/share/proj
make
make install
```

L'installation devrait se faire dans `c :\msys\local\grass-6.3.cvs`

Ces pages pourraient aussi être utiles :

– http ://grass.gdf-hannover.de/wiki/WinGRASS_Current_Status
– http ://geni.ath.cx/grass.html

G.2.4 GDAL étape deux

À cette étape, nous allons utiliser les sources de GDAL que nous avons utilisé précédemment, seule la compilation diffère.

Mais d'abord, vous devez pouvoir compiler GDAL avec le CVS actuel de GRASS en le fixant avec :

> http ://trac.osgeo.org/gdal/attachment/ticket/1587/plugin_patch_grass63.diff

(vous pouvez les fixer à la main ou avec patch.exe dans `c :\msys\bin`)

maintenant dans la console MSYS allez dans le répertoire des sources de GDAL et lancez les mêmes commandes que l'étape une, avec une différence :

```
1)rajoutez cet argument lors de ./configure:
--with-grass=/usr/local/grass-6.3.cvs

2) quand vous faites appel à g++ à la ligne 5 (ce qui crée libgdal.dll), ajoutez ces arguments:
-L/usr/local/grass-6.3.cvs/lib -lgrass\_vect -lgrass\_dig2 -lgrass\_dgl
-lgrass\_rtree \
-lgrass\_linkm -lgrass\_dbmiclient -lgrass\_dbmibase -lgrass\_I -lgrass\_gproj
\
-lgrass\_vask -lgrass\_gmath -lgrass\_gis -lgrass\_datetime}
```

Puis encore une fois éditez `gdal-config` et changez la ligne CONFIG_LIBS

```
CONFIG_LIBS="-L/usr/local/lib -lpng -L/usr/local/grass-6.3.cvs/lib -lgrass_vect
\
-lgrass_dig2 -lgrass_dgl -lgrass_rtree -lgrass_linkm -lgrass_dbmiclient \
-lgrass_dbmibase -lgrass_I -lgrass_gproj -lgrass_vask -lgrass_gmath -lgrass_gis
\
-lgrass_datetime -lz -L/usr/local/lib -lgdal"
```

Maintenant GDAL devrait fonctionner avec les couches vecteurs de GRASS.

G.2.5 GEOS

Téléchargez les sources depuis :

http://geos.refractions.net/geos-2.2.3.tar.bz2

Extrayez les vers `c :\msys\local\src`

Pour compiler, j'ai du changer le fichier `source/headers/timeval.h` à la ligne 13 :

```
#ifdef _WIN32
```

vers :

```
#if defined(_WIN32) && defined(_MSC_VER)
```

Dans la console MSYS, allez dans le répertoire source et faites :

```
./configure --prefix=/usr/local
make
make install
```

G.2.6 SQLITE

Vous pouvez utiliser la DLL précompilé en téléchargeant cette archive : http://www.sqlite.org/sqlitedll-3_3_17.zip

et copiez sqlite3.dll vers `c :\msys\local\lib`

Puis télécharger :

`http://www.sqlite.org/sqlite-source-3_3_17.zip`

et copiez sqlite3.h vers `c :\msys\local\include`

G.2.7 GSL

Sources à télécharger :

`ftp://ftp.gnu.org/gnu/gsl/gsl-1.9.tar.gz`

A extraire vers `c :\msys\local\src`

Allez dans le répertoire des sources via la console MSYS :

```
./configure
make
make install
```

G.2.8 EXPAT

Téléchargez les sources :

`http://dfn.dl.sourceforge.net/sourceforge/expat/expat-2.0.0.tar.gz`

Extrayez vers `c :\msys\local\src`

Allez dans le répertoire des sources via la console MSYS :

```
./configure
make
make install
```

G.2.9 POSTGRES

On va utiliser les binaires précompilés que vous pouvez télécharger ici :

```
http://wwwmaster.postgresql.org/download/mirrors-ftp?file=\%2Fbinary\%2Fv8.2.4\%
2Fwin32 \
\%2Fpostgresql-8.2.4-1-binaries-no-installer.zip
```

copiez le contenu du répertoire pgsql depuis l'archive vers `c :\msys\local`

G.3 Nettoyage

Nous avons terminé la création d'un environnement MSYS, vous pouvez effacer tout ce qui ce trouve dans `c :\msys\local\src`,ce n'est plus utile et gâche de l'espace disque.

H Compiler avec MS Visual Studio

/!\ Cette section décrit comment compiler soi-même toutes les dépendances. Lisez la prochaine section pour une procédure plus simple où toutes les dépendances sont déjà préparées.

Note : Cela n'inclut actuellement pas GRASS et Python.

H.1 Installer Visual Studio

Cette section décrit la configuration requise pour permettre à Visual Studio d'être utilisé pour compiler QGIS.

H.1.1 Express Edition

La version Express est gratuite mais ne dispose pas du SDK qui contient les en-têtes qui sont indispensables à la compilation de QGIS. Le SDK peut être installé comme décrit ici :

`http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/`

Une fois cela fait, vous devez éditez le fichier <vsinstalldir>\Common7\Tools\vsvars comme il suit :

> Ajoutez `%PlatformSDKDir%\Include\atl` et `%PlatformSDKDir%\Include\mfc` à l'entrée `@set INCLUDE`.

Cela ajoutera plus d'en-têtes dans le chemin INCLUDE du système. **Note :** Cela ne fonctionne que si vous utilisez la console de commandes quand vous compilez avec Visual Studio. Vous devez également faire une modification pour pallier l'absence d'une bibliothèque dans Visual Studio Express :

`http://www.codeproject.com/wtl/WTLExpress.asp`

H.1.2 Toutes les Éditions

Vous avez besoin de stdint.h et unistd.h. unistd.h vient avec la version GnuWin32 des binaires de flex & bison (à voir par la suite). stdint.h peut se trouver ici :

```
http://www.azillionmonkeys.com/qed/pstdint.h
```

Copiez les deux vers <vsinstalldir>\VC\include.

H.2 Télécharger/Installer les Dépendances

Cette section décrit le téléchargement et l'installation des différentes dépendances de QGIS.

H.2.1 Flex et Bison

Télécharger les paquets suivants et lancez les installateurs :

```
http://gnuwin32.sourceforge.net/downlinks/flex.php
http://gnuwin32.sourceforge.net/downlinks/bison.php
```

H.2.2 Pour inclure le support de PostgreSQL dans Qt

Vous devez télécharger PostgreSQL, l'installer et créer une bibliothèque que vous pourrez ensuite lier à Qt.

Téléchargez .../binary/v8.2.5/win32/postgresql-8.2.5-1.zip depuis un serveur de PostgreSQL.org.

PostgreSQL est actuellement compilé avec MinGW et dispose d'en-têtes et une bibliothèque pour MinGW. Les en-têtes peuvent être utilisés directement avec Visual C++ mais la bibliothèque est seulement fournit sous forme DLL ou archive (.a) et ne peut donc pas être utilisés directement avec Visual C++. Visual C++ directly.

Pour créer une bibliothèque, copiez le script sed suivant dans le fichier mkdef.sed dans répertoire lib de PostgreSQL :

```
/Dump of file / {
 s/Dump of file \([^  ]*\)$/LIBRARY \1/p
 a\
EXPORTS
}
/[  ]*ordinal hint/,/^[ ]*Summary/ {
```

```
 /^[  ]\+[0-9]\+/ {
   s/^[  ]\+[0-9]\+[  ]\+[0-9A-Fa-f]\+[  ]\+[0-9A-Fa-f]\+[
]\+\([^  =]\+\).*$/ \1/p
 }
}
```

et éxucter ces commandes dans la console de Visual Studio C++ :

```
cd c:\Program Files\PostgreSQL\8.2\bin
dumpbin /exports ..\bin\libpq.dll | sed -nf ../lib/mkdef.sed >..\lib\libpq.def
cd ..\lib
lib /def:libpq.def /machine:x86
```

Vous aurez besoin de sed pour que ça marche (e.g. venant de cygwin ou de msys).

On y est presque. Vous devez modifier les chemins INCLUDE et LIB dans vcvars.bat.

H.2.3 Qt

Compiler Qt avec ces instructions :

http://wiki.qgis.org/qgiswiki/Building_QT_4_with_Visual_C%2B%2B_2005

H.2.4 Proj.4

Téléchargez les sources de proj.4 source depuis :

http://proj.maptools.org/

Avec la commande de Visual Studio, exécutez ce qui suit dans le répertoire src :

```
nmake -f makefile.vc
```

Installez en lançant les commandes suivantes dans le répertoire supérieur pour configurer correctement PROJ_DIR :

```
set PROJ_DIR=c:\lib\proj

mkdir %PROJ_DIR%\bin
mkdir %PROJ_DIR%\include
```

```
mkdir %PROJ_DIR%\lib

copy src\*.dll %PROJ_DIR%\bin
copy src\*.exe %PROJ_DIR%\bin
copy src\*.h %PROJ_DIR%\include
copy src\*.lib %PROJ_DIR%\lib
```

Vous pouvez rassembler ces commandes dans un script.

H.2.5 GSL

À télécharger depuis :

http://david.geldreich.free.fr/downloads/gsl-1.9-windows-sources.zip

Compilez en utilisant le fichier gsl.sln.

H.2.6 GEOS

Téléchargez geos depuis svn (svn checkout http ://svn.refractions.net/geos/trunk geos). Editez geos\source\makefile.vc comme il suit :

Décommentez les lignes 333 et 334 pour permettre de copier de la version.h.vc vers la version.h.

Décommentez les lignes 338 et 339.

Renommez geos_c.h.vc vers geos_c.h.in aux lignes 338 and 339 pour permettre de copier geos_-c.h.in vers geos_c.h.

Avec la commande de Visual Studio (assurez que l'environnement est bien configuré), lancez ça dans le répertoire supérieur :

```
nmake -f makefile.vc
```

Lancez les commandes suivantes dans le répertoire supérieur pour configurer correctement GEOS_-DIR :

```
set GEOS_DIR="c:\lib\geos"

mkdir %GEOS_DIR%\include
mkdir %GEOS_DIR%\lib
```

```
mkdir %GEOS_DIR%\bin

xcopy /S/Y source\headers\*.h %GEOS_DIR%\include
copy /Y capi\*.h %GEOS_DIR%\include
copy /Y source\*.lib %GEOS_DIR%\lib
copy /Y source\*.dll %GEOS_DIR%\bin
```

H.2.7 GDAL

Téléchargez gdal depuis svn(svn checkout https ://svn.osgeo.org/gdal/branches/1.4/gdal gdal).

Éditez nmake.opt pour qu'il s'intègre correctement, le tout est assez bien commenté.

Avec la commande de Visual Studio (assurez que l'environnement est bien configuré), lancez ça dans le répertoire supérieur :

```
nmake -f makefile.vc
```

et

```
nmake -f makefile.vc devinstall
```

H.2.8 PostGIS

Télécharger la version Windows de PostgreSQl et PostGIS ici :

`http://postgis.refractions.net/download/`

Note : the warning about not installing the version of PostGIS that comes with the PostgreSQL installer. Simply run the installers.

H.2.9 Expat

Téléchargez Expat depuis :

`http://sourceforge.net/project/showfiles.php?group_id=10127`

Vous avez besoin de expat-win32bin-2.0.1.exe.

Lancez l'exécutable pour installer expat.

H.2.10 CMake

Télécharger CMake depuis :

http://www.cmake.org/HTML/Download.html

Vous avez besoin de cmake-<version>-win32-x86.exe. Lancez-le pour installer CMake.

H.3 Building QGIS with CMAKE

Téléchargez les sources QGIS depuis svn (svn co https ://svn.osgeo.org/qgis/trunk/qgis qgis).

Créer un répertoire 'Build' dans le dossier supérieur de QGIS pour accueillir le résultat de la compilation.

Cliquez sur Démarrer–>Tout les programmes–>CMake–>CMake.

Dans la boîte 'Where is the source code :', allez jusqu'au répertoire supérieur de QGIS.

Dans la boîte 'Where to build the binaries :', allez dans le répertoire 'Build' que vous avez créé.

Remplissez les différentes entrées *_INCLUDE_DIR et *_LIBRARY dans la liste 'Cache Values'.

Cliquez le bouton Configure. Il vous est demandé quel type de makefile va être généré, sélectionnez Visual Studio 8 2005 et cliquez OK.

Si tout va bien la compilation devrait se faire sans erreurs. Si il y en a, c'esthabituellement dû à un chemin incorrect vers le répertoire de l'en-tête ou de la bibliothèque. Les objets qui ont échoué sont marqués en rouge.

Une fois la configuration correctement terminée, cliquez sur OK pour générer les fichiers de projet.

Avec Visual Studio 2005, ouvrez le fichier qgis.sln qui a été créé dans le répertoire 'Build'.

Compiler le projet ALL_BUILD. Cela compilera tout les binaires QGIS ainsi que toutes les extensions.

Installez QGIS en lançant le projet INSTALL. Par défaut, l'installation se fera dans c :\Program Files\qgis<version> (ce qui peut être changé avec la variable CMAKE_INSTALL_PREFIX de CMake).

Vous allez aussi avoir besoin d'ajouter toutes les DLLs des dépendances dans le répertoire d'installation de QGIS ou bien ajouter leurs répertoires respectifs dans votre chemin (PATH).

I Compiler sous Windows avec MSVC Express

Note : : La compilation sous Windows à l'aide de MSVC Express est un travail en cours. En particulier, les choses suivantes ne fonctionnent pas : python, grass, postgis connections. / !\ Cette partie de document est à l'état de brouillon et ne peut donc pas encore être utilisé.

Tim Sutton, 2007

I.1 Préparation du système

J'ai commencé avec une installation récente de Windows XP avec le Service Pack 2 et tous les correctifs appliqués. J'ai déjà compilé toutes les dépendances nécessaires pour gdal, expat etc, donc ce guide ne couvrira également pas leurs compilation depuis les sources. Du fait de la difficulté de les compiler, j'espère que mes binaires vous seront utiles. Dans le cas contraire, je vous conseille de consulter les documentations spécifiques des différents projets. Résumons rapidement le processus avant de commencer :

* Installer XP * Installer les binaires que j'ai fait * Installer Visual Studio Express 2005 sp1 * Installer le Microsoft Platform SDK * Installer le client console de SVN * Installer l'ensemble des dépendances * Installer Qt 4.3.2 * obtenir les sources de QGIS * Compiler QGIS * Créer un installateur setup.exe pour QGIS

I.2 Installer les bibliothèques

Cette procédure a essentiellement pour but de vous simplifier les choses. A cette fin, j'ai créé une archive qui inclue toutes les dépendances nécessaires pour compiler QGIS à l'exception de Qt (que nous compilerons par la suite). Récupérer ce fichier à :

```
http://qgis.org/uploadfiles/msvc/qgis_msvc_deps_except_qt4.zip
```

Créez cette structure de répertoire :

```
c:\dev\cpp\
```

Extrayez l'archive dans un sous-répertoire de manière à avoir :

```
c:\dev\cpp\qgislibs-release
```

Note : vous n'êtes pas obligés se suivre cette structure, mais dans ce cas vous devrez adapté les instructions qui suivent.

I.3 Installer Visual Studio Express 2005

la première chose à faire est de télécharger MSVC Express depuis :

http ://msdn2.microsoft.com/en-us/express/aa975050.aspx

Voici les options que j'ai choisies :

* Envoyer les informations d'usage à Microsoft (Non) * Options d'installation : * IDE graphique (Oui) * Microsoft MSDN Express Edition (Non) * Microsoft SQL Server Express Edition (Non) * Installation vers le dossier C :\Program Files\Microsoft Visual Studio 8\ (default)

I.4 Installer Microsoft Platform SDK2

Allez à cette page :

http ://msdn2.microsoft.com/en-us/express/aa700755.aspx

Cherchez ces 3 fichiers et sélectionnez celui qui correspond à votre plateforme :

```
PSDK-amd64.exe  1.2 MB  Download
PSDK-ia64.exe   1.3 MB  Download
PSDK-x86.exe    1.2 MB  Download
```

Faites une installation personnalisée et installez le logiciel vers :

```
C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\
```

Nous allons faire une installation minimale pour avoir un environnement fonctionnel :

```
Configuration Options
  + Register Environmental Variables            (Oui)
Microsoft Windows Core SDK
  + Tools                                       (Oui)
    + Tools (AMD 64 Bit)                        (Non à moins que ce soit le cas)
    + Tools (Intel 64 Bit)                      (Non à moins que ce soit le cas)
  + Build Environment
    + Build Environment (AMD 64 Bit)            (Non à moins que ce soit le cas)
    + Build Environment (Intel 64 Bit)          (Non à moins que ce soit le cas)
    + Build Environment (x86 32 Bit)            (Oui)
  + Documentation                               (Non)
```

```
  + Redistributable Components                (Oui)
  + Sample Code                               (Non)
  + Source Code                               (Non)
    + AMD 64 Source                           (Non)
    + Intel 64 Source                         (Non)
Microsoft Web Workshop                        (Oui) (nécessaire pour shlwapi.h)
  + Build Environment                         (Oui)
  + Documentation                             (Non)
  + Sample Code                               (Non)
  + Tools                                     (Non)
Microsoft Internet Information Server (IIS) SDK (Non)
Microsoft Data Access Services (MDAC) SDK     (Oui) (nécessaire pour GDAL/odbc)
  + Tools
    + Tools (AMD 64 Bit)                      (Non)
    + Tools (AMD 64 Bit)                      (Non)
    + Tools (x86 32 Bit)                      (Oui)
  + Build Environment
    + Tools (AMD 64 Bit)                      (Non)
    + Tools (AMD 64 Bit)                      (Non)
    + Tools (x86 32 Bit)                      (Oui)
  + Documentation                             (Non)
  + Sample Code                               (Non)
Microsodt Installer SDK                       (Non)
Microsoft Table PC SDK                        (Non)
Microsoft Windows Management Instrumentation  (Non)
Microsoft DirectShow SDK                      (Non)
Microsoft Media Services SDK                  (Non)
Debuggin Tools for Windows                    (Oui)
```

Note : vous pourrez toujours rajouter des éléments par la suite.

Note : l'installation du SDK requiert la validation du Microsoft Genuine Advantage application. Si vous ne pouvez ou ne voulez pas, il vous faudra recourir aux instructions de compilations avec MINGW de ce document.

Le SDK créé un répertoire nommé

```
C:\Office10
```

Que vous pouvez effacer sans crainte.

Après son installation, suivez les notes restantes sur la page indiquée précédemment pour finir la

configuration de votre environnement MSVC Express. Pour votre confort, voici un bref résumé de ces étapes avec quelques autres détails :

1) Ouvrez Visual Studio Express IDE

2) Tools -> Options -> Projects and Solutions -> VC++ Directories

3) Ajoutez :

```
Executable files:
  C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Bin

Include files:
  C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include
  C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include\atl
  C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include\mfc
Library files: C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Lib
```

4) Fermer MSVC Express IDE

5) Ouvrez le fichier suivant avec notepad :

```
C:\Program Files\Microsoft Visual Studio
8\VC\VCProjectDefaults\corewin_express.vsprops
```

et changez la propriété :

```
AdditionalDependencies="kernel32.lib"
```

Pour :

```
AdditionalDependencies="kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib
                        advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib"
```

I.5 Editer vos vsvars

Faites une sauvegarde de votre fichier vsvars32.bat présent dans

```
C:\Program Files\Microsoft Visual Studio 8\Common7\Tools
```

et remplacez le par :

```
@SET VSINSTALLDIR=C:\Program Files\Microsoft Visual Studio 8
@SET VCINSTALLDIR=C:\Program Files\Microsoft Visual Studio 8\VC
@SET FrameworkDir=C:\WINDOWS\Microsoft.NET\Framework
@SET FrameworkVersion=v2.0.50727
@SET FrameworkSDKDir=C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0
@if "%VSINSTALLDIR%"=="" goto error_no_VSINSTALLDIR
@if "%VCINSTALLDIR%"=="" goto error_no_VCINSTALLDIR

@echo Setting environment for using Microsoft Visual Studio 2005 x86 tools.

@rem
@rem Root of Visual Studio IDE installed files.
@rem
@set DevEnvDir=C:\Program Files\Microsoft Visual Studio 8\Common7\IDE

@set PATH=C:\Program Files\Microsoft Visual Studio 8\Common7\IDE;C:\Program \
Files\Microsoft Visual Studio 8\VC\BIN;C:\Program Files\Microsoft Visual Studio
8\ \
Common7\Tools;C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\bin; \
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727;C:\Program Files\Microsoft Visual
\
Studio 8\VC\VCPackages;%PATH%
@rem added by Tim
@set PATH=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
R2\Bin;%PATH%
@set INCLUDE=C:\Program Files\Microsoft Visual Studio 8\VC\INCLUDE; \
%INCLUDE%
@rem added by Tim
@set INCLUDE=C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\
\
Include;%INCLUDE%
@set INCLUDE=C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\
\
Include\mfc;%INCLUDE%
@set INCLUDE=%INCLUDE%;C:\dev\cpp\qgislibs-release\include\postgresql
@set LIB=C:\Program Files\Microsoft Visual Studio 8\ \
VC\LIB;C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\lib;%LIB%
@rem added by Tim
@set LIB=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
R2\Lib;%LIB%
```

```
@set LIB=%LIB%;C:\dev\cpp\qgislibs-release\lib
@set LIBPATH=C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727

@goto end

:error_no_VSINSTALLDIR
@echo ERROR: VSINSTALLDIR variable is not set.
@goto end

:error_no_VCINSTALLDIR
@echo ERROR: VCINSTALLDIR variable is not set.
@goto end

:end
```

I.6 Variables d'environnement

Clic-droit sur 'Poste de travail' puis sélectionnez le panneau 'Avancé'. Cliquez sur les variables d'environnement et créez ou ouvrez les variables "'Système"' :

```
Variable Name:     Value:
--------------------------------------------------------------------------
EDITOR             vim
INCLUDE            C:\Program Files\Microsoft Platform SDK for Windows Server
2003 R2 \
\Include\.
LIB                C:\Program Files\Microsoft Platform SDK for Windows Server
2003 R2 \
\Lib\.
LIB_DIR            C:\dev\cpp\qgislibs-release
PATH               C:\Program Files\CMake 2.4\bin;
                   %SystemRoot%\system32;
                   %SystemRoot%;
                   %SystemRoot%\System32\Wbem;
                   C:\Program Files\Microsoft Platform SDK for Windows Server
2003 R2 \
                   \Bin\.;
                   C:\Program Files\Microsoft Platform SDK for Windows Server
2003 R2\  \
                   \Bin\WinNT\;
```

```
                    C:\Program Files\svn\bin;C:\Program Files\Microsoft Visual
Studio 8 \
                    \VC\bin;
                    C:\Program Files\Microsoft Visual Studio 8\Common7\IDE;
                    "c:\Program Files\Microsoft Visual Studio 8\Common7\Tools";
                    c:\Qt\4.3.2\bin;
                    "C:\Program Files\PuTTY"
QTDIR               c:\Qt\4.3.2
SVN_SSH             "C:\\Program Files\\PuTTY\\plink.exe"
```

I.7 Compiler Qt4.3.2

Qt 4.3.2 est la version miniale requise puisque c'est la première version à supporter la compilation la version Open source de Qt avec MSVC.

Téléchargez les sources Qt 4.x.x depuis :

```
http:\\www.trolltech.com
```

Extrayez les sources vers

```
c:\Qt\4.x.x\
```

I.7.1 Compiler Qt

Ouvrez la ligne de commande de Visual Studio C++ et déplacez-vous vers c :\Qt\4.x.x et entrez :

```
configure -platform win32-msvc2005
nmake
nmake install
```

Ajoutez -qt-sql-odbc -qt-sql-psql dans ligne 'configure' si vous voulez le support odbc et PostgreSQL dans Qt.

Note :J'ai eu quelques erreurs avec qscreenshot.pro. Si vous ne voulez que compiler des applications Qt vous pouvez l'ignorer. Vérifiez dans c :\Qt\4.3.2\bin que toutes les DLLs et les aides (assistant, etc.) ont été créés.

I.7.2 Configuration de Visual C++ pour utiliser Qt

Après la compilation, configurez Visual Studio Express IDE pour utiliser Qt :

1) ouvrez Visual Studio Express IDE

2) Tools -> Options -> Projects and Solutions -> VC++ Directories

3) Ajoutez :

```
Executable files:
  $(QTDIR)\bin

Include files:
  $(QTDIR)\include
  $(QTDIR)\include\Qt
  $(QTDIR)\include\QtCore
  $(QTDIR)\include\QtGui
  $(QTDIR)\include\QtNetwork
  $(QTDIR)\include\QtSvg
  $(QTDIR)\include\QtXml
  $(QTDIR)\include\Qt3Support
  $(LIB_DIR)\include   (needed during qgis compile to find stdint.h and
unistd.h)

Library files:
  $(QTDIR)\lib

Source Files:
  $(QTDIR)\src
```

Hint : You can also add

```
QString = t=<d->data, su>, size=<d->size, i>
```

to AutoExp.DAT in C :\Program Files\Microsoft Visual Studio 8\Common7\Packages\Debugger before

```
[Visualizer]
```

Comme cela, le debugger montrera le contenu de QString quand vous en aurez besoin. On peut aussi ajouter d'autres choses (faites le si vous en avez besoin) j'avais juste besoin de QString et j'ai pris la première occurence que j'ai pu trouver dans google

I.8 Installer Python

Télécharger http ://python.org/ftp/python/2.5.1/python-2.5.1.msi et installez-le.

I.9 Installer SIP

Téléchargez http ://www.riverbankcomputing.com/Downloads/sip4/sip-4.7.1.zipet extrayez le dans votre répertoire c :\dev\cpp. Depuis la console de Visual C++, déplacez vous dans ce dossier et faites :

```
c:\python25\python configure.py -p win32-msvc2005
nmake
nmake install
```

I.10 Installer PyQt4

Téléchargez http ://www.riverbankcomputing.com/Downloads/PyQt4/GPL/PyQt-win-gpl-4.3.1.zip et extrayez-le dans votre répertoire c :\dev\cpp. Depuis la console de Visual C++, déplacez-vous dans ce dossier et faites :

```
c:\python25\python configure.py -p win32-msvc2005
nmake
nmake install
```

I.11 Installer CMake

Téléchargez et installez cmake 2.4.7 ou plus récent, assurez-vous d'activer cette option : `Update path for all users`

I.12 Installer Subversion

Vous "'devez"' installer la version en ligne de commande si vous voulez que les scripts CMake pour svn fonctionnent. Il est difficile de trouver la version correcte sur le site du projet subversion du fait des noms parfois similaires, dirigez-vous plutôt vers ce fichier :

http ://subversion.tigris.org/downloads/1.4.5-win32/apache-2.2/svn-win32-1.4.5.zi p

Extrayez le fichier zip vers

```
C:\Program Files\svn
```

Et ajoutez

```
C:\Program Files\svn\bin
```

à votre chemin.

I.13 Récupération SVN initiale

Ouvrer une fenêtre cmd.exe et faites :

```
cd \
cd dev
cd cpp
svn co https://svn.osgeo.org/qgis/trunk/qgis
```

A ce point, vous allez sûrement avoir un message comme celui-ci :

```
C:\dev\cpp>svn co https://svn.osgeo.org/qgis/trunk/qgis
Error validating server certificate for 'https://svn.qgis.org:443':
 - The certificate is not issued by a trusted authority. Use the
   fingerprint to validate the certificate manually!
Certificate information:
 - Hostname: svn.qgis.org
 - Valid: from Sat, 01 Apr 2006 03:30:47 GMT until Fri, 21 Mar 2008 03:30:47 GMT
 - Issuer: Developer Team, Quantum GIS, Anchorage, Alaska, US
 - Fingerprint: 2f:cd:f1:5a:c7:64:da:2b:d1:34:a5:20:c6:15:67:28:33:ea:7a:9b
(R)eject, accept (t)emporarily or accept (p)ermanently?
```

Appuyez sur 'p' pour accepter et la récupération svn devrait débuter.

I.14 Créer des Makefiles avec cmakesetup.exe

Je ne vais pas donner une description détaillée du processus de compilation car il est déjà expliqué dans la première partie de ce guide. Sautez les étapes sur la compilation de GDAL etc., puisque ce processus d'installation simplifié prend en charge les dépendances à votre place.

```
cd qgis
mkdir build
cd build
cmakesetup ..
```

Cmakesetup devrait récupérer toutes les dépendances pour vous, et ce automatiquement (il utilise l'environnement LIB_DIR pour les repérer dans c :\dev\cpp\qgislibs-release). Appuyez encore sur configurer après l'apparition de l'interface de cmakesetup et quand tous les champs en rouges ont disparu, appuyez sur OK pour fermer cmake.

Maintenat ouvrez Visual Studio Express et faites : `File -> Open -> Project /Solution`

Puis ouvrez le fichier QGIS généré par cmake qui devrait être dans :

```
c:\dev\cpp\qgis\build\qgisX.X.X.sln
```

Où X.X.X représente la version actuelle de QGIS. Pour le moment j'ai seulement distribué les dépendances pour QGIS, les versions de débug devraient suivre, vous devez donc sélectionner 'Release' dans la barre de configuration. Ensuite, faites un clic droit sur ALL_BUILD dans le navigateur et choisissez 'build'. Une fois la compilation achevée, faites un clic droit sur INSTALL dans le navigateur et choisissez 'build'. Cela installera par défaut QGIS dans c :\program files\qgisX.X.X.

I.15 Executer et empaqueter

Pour lancer QGIS vous aurez besoin d'une copie des DLLs de c :\dev\cpp\qgislibs-release\bin vers c :\program files\qgisX.X.X directory.

J Standards de Codage de QGIS

Le chapitre suivant fournie des informations de codage pour la version 1.0.0 de QGIS. Ce document correspond quasiment à une conversion en LATEXdu fichier CODING.txt fournit avec les sources de QGIS du 16 décembre 2008.

Ces standards doivent être suivis par tous les développeurs de QGIS. Les informations actuelles sur les standards de codage de QGIS sont également disponibles sur le wiki :

http://wiki.qgis.org/qgiswiki/CodingGuidelines
http://wiki.qgis.org/qgiswiki/CodingStandards
http://wiki.qgis.org/qgiswiki/UsingSubversion
http://wiki.qgis.org/qgiswiki/DebuggingPlugins
http://wiki.qgis.org/qgiswiki/DevelopmentInBranches
http://wiki.qgis.org/qgiswiki/SubmittingPatchesAndSvnAccess

J.1 Classes

J.1.1 Noms

Les classes dans QGIS débutent avec Qgs et sont formées en utilisant une casse mixte.

```
Exemples :
QgsPoint
QgsMapCanvas
QgsRasterLayer
```

J.1.2 Membres

Les noms des membres de classe débutent avec un *m* en minuscule et sont formés de casses mixtes.

```
mMapCanvas
mCurrentExtent
```

Tous les membres de classes doivent être privés.

Les membres publics de classes sont fortement déconseillés.

J.1.3 Fonctions accesseurs

Les membres des classes doivent être obtenus par une fonction accesseurs. La fonction doit être nommée avec un préfixe *get*. les fonctions accesseurs pour deux membres privées doivent être :

```
mapCanvas()
currentExtent()
```

J.1.4 Fonctions

Les noms de fonction doivent commencer par une lettre en minuscule et sont formés de casse mixte. Le nom de la fonction doit se rapporter à l'objectif de la fonction.

```
updateMapExtent()
setUserOptions()
```

J.2 Qt Designer

J.2.1 Classes générées

Les classes de QGIS qui sont générées à partir des fichiers de Qt Designer (ui) doivent avoir un suffixe *Base*. Cela identifie la classe comme classe de base générée.

```
Exemples :
QgsPluginMangerBase
QgsUserOptionsBase
```

J.2.2 Boîtes de dialogues

Toutes les boîtes de dialogue implémentent ce qui suit : * aide en tootlip pour toutes les icônes de la barre d'outils et autre widgets pertinents ; * aide WhatsThis pour **tous** les widgets sur la boîte de dialogue ; * un bouton *Help* fonction du contexte optionnel (bien que fortement recommandé) qui redirige l'utilisateur vers une page d'aide appropriée en lançant leur navigateur web.

J.3 Files C++

J.3.1 Noms

Les fichiers d'implémentations C++ et d'en-tête doivent avoir respectivement une extension.cpp et .h. Les noms de fichier doivent être en minuscule et dans le cas des classes, correspondre au nom de la classe.

```
Exemple :
Class QgsFeatureAttribute source files are
qgsfeatureattribute.cpp and qgsfeatureattribute.h
```

J.3.2 En-tête standard et license

Chaque fichier source doit contenir un section d'en-tête sur le modèle de cet exemple :

```
/***************************************************************************
    qgsfield.cpp - Describes a field in a layer or table
     --------------------------------------
    Date                 : 01-Jan-2004
    Copyright            : (C) 2004 by Gary E.Sherman
    Email                : sherman at mrcc.com
/***************************************************************************
 *                                                                         *
 *   This program is free software; you can redistribute it and/or modify  *
 *   it under the terms of the GNU General Public License as published by  *
 *   the Free Software Foundation; either version 2 of the License, or     *
 *   (at your option) any later version.                                   *
 *                                                                         *
 ***************************************************************************/
```

J.3.3 Mot-clé CVS

Chaque fichier source doit contenir le mot-clé Id. Cela sera remplacé par CVS pour contenir des informations utiles sur le fichier, la révision, le dernier commiteur et la date et l'heure de la dernière récupération. Placer le mot-clé directement après la licence/l'en-tête standard qui est trouvé en haut de chaque fichier source :

```
/* $Id$ */
```

J.4 Noms de variables

Les noms de variables débutent avec une minuscule et sont formés de casse mixte.

```
Exemples :
mapCanvas
currentExtent
```

J.5 Types d'énumération

Les types d'énumération doivent être nommés en CamelCase avec la première lettre en capital, par exemple :

```
enum UnitType
{
  Meters,
  Feet,
  Degrees,
  UnknownUnit
} ;
```

N'utilisez pas de nom de type générique qui entrera en conflit avec d'autres types. Par exemple, utiliser “UnkownUnit” plutôt que “Unknown”.

J.6 Constantes globales

Les constantes globales doivent être écrit en majuscule séparée par des symboles soulignement, par exemple :

```
const long GEOCRS_ID = 3344;
```

J.7 Édition

N'importe quel éditeur/IDE peut être utilisé pour éditer le code de QGIS s'il fournit les besoins suivants :

J.7.1 Tabulations

Définissez votre éditeur pour simuler les tabulations avec des espaces. L'espace des tabulations doit être de 2 espaces.

J.7.2 Indentation

Le code source doit être indenté pour améliorer la lisibilité. Il y a un fichier .ident.pro dans le répertoire src de QGIS qui contient la correspondance à utiliser pour indenter le code en utilisant le programme GNU indent. Si vous n'utilisez pas ce programme, vous devez simuler ces paramétrages.

J.7.3 Accolades

Les accolades débutent sur la ligne suivant l'expression :

```
if(foo == 1)
{
  // faire quelque chose
  ...
  }else
{
  // faire quelque chose d'autre
  ...
}
```

J.8 Compatibilitie avec l'API

À partir de QGIS 1.0, nous fournissons une API stable et compatible arrière. Ceci fournira une base stable sur laquelle les personnes développeront, en sachant que leur code restera compatible avec les versions 1.x de QGIS (bien qu'une recompilation puisse être nécessaire). Le nettoyage de l'API sera effectué de la même manière que les développeurs de Trolltech, par exemple :

```
class Foo
{
  public:
    /** This method will be deprecated, you are encouraged to use
      doSomethingBetter() rather.
      @see doSomethingBetter()
```

```
    */
   bool doSomething();

   /** Does something a better way.
     @note This method was introduced in QGIS version 1.1
    */
   bool doSomethingBetter();

}
```

J.9 Style de codage

Voici la description de quelques idées et astuces de programmation qui permettra de réduire les erreurs, le temps de développement et de maintenance.

J.9.1 Généraliser le code quand c'est possible

Si vous copiez collé du code ou du moins, recopiez la même portion de code, considérez la généralisation du code en une fonction.

Cela permettra de modifier en un seul endroit au lieu d'en plusieurs endroits.

– aide à prévenir l'hypertrophie du code ;
– rend plus difficile l'évolution dans le temps de multiple copies de codes et donc complique la compréhension et le maintient pour les autres.

J.9.2 Préférer les constantes en premier dans les prédicats

Préférez le placement des constants en début dans les prédicats.

```
"0 == value" au lieu de "value == 0"
```

Cela aidera à prévenir les utilisations accidentelles de "=" au lieu d'utiliser "==" qui peut introduire des bugs subtils. Le compilateur rejettera une erreur si vous utilisez accidentellement "=" au lieu de "==" pour la comparaison puisqu'on ne peut pas assigné une valeur aux constantes.

J.9.3 Les espaces sont vos amis

Ajouter des espaces entre les opérateurs, commandes et fonctions permet de faciliter la lecture du code.

Lequel est plus facile à lire, ceci :

```
if (!a&&b)
```

ou ceci :

```
if ( ! a && b )
```

J.9.4 Ajouter des commantaires

Ajouter des commentaires à la fin des fonctions, des implémentations des structures et des classes, il est plus facile de les retrouver plus tard.

Considérez que vous êtes au fond d'un fichier source et que vous devez retrouver une fonction très longue - sans ce type de commentaires, vous devrez remonter dans le code pour retrouver son nom. Bien sûr, cela est ok si vous voulez trouver le début de la fonction, mais si vous vous intéressez au code près de la fin ? Vous devriez descendre puis parvenir à nouveau à la partie désirée.

Par exemple,

```
void foo::bar()
{
    // ... imaginez beacuoup de code ici
 } // foo::bar()
```

J.9.5 Utiliser des accolades même pour des commandes sur une seule ligne

En utilisant des accolades pour le code dans les blocs if/then ou des structures similaires, même pour une seule ligne de code signifie que l'ajout d'une autre déclaration est moins susceptible de générer un code cassé.

Considérez :

```
  if (foo)
    bar();
```

```
else
   baz();
```

Ajouter du code après bar() ou baz() sans ajouter d'accolades autour créera un code cassé. Bien que la plupart des programmeurs le feront naturellement, certains pourraient oublier.

Ainsi, préférez ceci :

```
if (foo)
{
  bar();
}
else
{
   baz();
}
```

J.9.6 Recommendations de livres

Effective C++ http://www.awprofessional.com/title/0321334876
More Effective C++ http://www.awprofessional.com/bookstore/product.asp?isbn=020163371X&rl=1
Effective STL http://www.awprofessional.com/title/0201749629
Design Patterns http://www.awprofessional.com/title/0201634988

Vous devez aussi vraiment lire cette article de la publication trimestrielle de Qt http://doc.trolltech.com/qq/qq13-apis.html

K Accès SVN

Cette page explicite la procédure de démarrage exploitant le répertoire QGIS Subversion.

K.1 Accéder au répertoire

Pour vérifier le QGIS HEAD :

```
svn --username [votre nom d'utilisateur] co https://svn.qgis.org/repos/qgis/trunk/qgis
```

K.2 Accès anonyme

Vous pouvez utiliser les commandes suivantes pour fournir un accès anonyme au dossier QGIS Subversion. Notez que nous recommandons de contrôler les troncs (sauf si vous êtes un développeur ou que vous avez vraiment BESOIN d'avoir les dernières modifications et que vous ne craignez pas de nombreux crashs).

Vous devez avoir une sous-version client installée avant de saisir le code. Regardez sur le site internet dédié aux Subversions pour plus d'information. La page des liens contient une sélection de clients SVN pour de nombreuses plateformes.

Pour vérifier une branche :

```
svn co https://svn.qgis.org/repos/qgis/branches/<branch name>
```

Pour vérifier le tronc :

```
svn co https://svn.qgis.org/repos/qgis/trunk/qgis qgis_unstable
```

/ !\ **Note :** Si vous êtes derrière un serveur proxy, éditez votre configuration proxy en premier.

/ !\ **Note :** Dans QGIS, nous conservons notre code le plus stable dans le tronc. Nous mettons à régulièrement une version en dehors du tronc, et ensuite nous continuons à stabiliser et incorporer une sélection de nouvelles fonctionnalités dans le tronc.

Veuillez regarder le fichier INSTALL à la racine de l'arbre pour les instructions spécifiques sur la construction et le développement des versions.

K.3 Documentation des sources de QGIS

Si vous souhaitez étudier la documentation des sources de Quantum GIS :

```
svn co https://svn.qgis.org/repos/qgis_docs/trunk qgis_docs
```

Vous pouvez de même regarder la section « DocumentationWritersCorner » pour plus d'information.

K.4 Documentation

Le répertoire est organisé comme suit :

http ://wiki.qgis.org/images/repo.png

Veuillez vous reporter au livre des Subversion http ://svnbook.red-bean.com pour obtenir de plus ample information pour devenir un SVN master.

K.5 Développement des branches

K.5.1 Sujet

Le code de QGIS s'est considérablement complexifié ces dernières années. Ainsi, il est désormais dur d'anticiper les effets de bords que l'addition d'une fonctionnalité peut avoir. Dans le passé, les cycles développement des versions du projet QGIS furent très longs, car d'importants travaux pour ré-établir la stabilité du logiciel furent nécessaires après l'ajout de nouvelles composantes. Pour venir à bout de ces problèmes, QGIS choisit un modelé de développement dans lequel les nouvelles fonctionnalités sont codées dans des branches du SVN et fusionnées dans le tronc (la branche principale) quand elles sont finies et stables. Cette section décrit la procédure allant de la branche à la fusion dans le projet QGIS.

K.5.2 Procédure

* Annonce initiale dans la liste de discussion Avant de commencer, veuillez poster une annonce sur la liste de discussion des développeurs pour voir si un autre développeur n'est pas déjà en train de travailler sur la même fonctionnalité. Veuillez aussi contacter le responsable technique (technical advisor) du comité pilotage du projet (project steering committee — PSC). Si la nouvelle fonctionnalité demande des changements dans l'architecture QGIS, une requête pour commentaire (request for comment — RFC) est nécessaire. * Créer une nouvelle branche Créez une nouvelle branche dans le SVN correspondant à la nouvelle fonctionnalité (voir UsingSubversion pour la syntaxe SVN). Vous pouvez désormais commencer le développement. * Fusionner régulièrement à partir du tronc Il est recommandé de fusionner périodiquement les changements effectués dans le tronc avec la branche. Ces fusions facilitent la fusion ultérieure de la branche avec le tronc.* Documenter sur le wiki Il est de même conseillé de documenter les changements prévus ainsi que l'état du travail en cours sur une page du wiki. * Tester avant de fusionner dans la branche principale Quand vous avez fini le développement de la nouvelle fonctionnalité et satisfait de sa stabilité, veuillez poster une annonce sur la liste de discussion des développeurs. Avant de fusionner avec le tronc, les changements doivent être testés par les développeurs et les utilisateurs. Les versions binaires (surtout pour OsX et Windows) seront générées pour impliquer aussi les non-développeurs. Dans le suivi (le trac), un nouveau composant sera ouvert contre le ticket du fichier. Dès qu'il n'y a plus de question en suspens, le responsable technique du PSC fusionne les changements dans le tronc.

K.5.3 Créer une branche

Nous préférons que le développement d'une nouvelle fonctionnalité se fasse en dehors du tronc, donc que le tronc reste à un état stable. Pour créer une branche, veuillez utiliser les commandes suivantes :

```
svn copy https://svn.qgis.org/repos/qgis/trunk/qgis \
https://svn.qgis.org/repos/qgis/branches/qgis_newfeature
svn commit -m "New feature branch"
```

K.5.4 Fusionner régulièrement le tronc dans la branche

Quand vous travaillez dans une branche, vous devez régulièrement fusionner le tronc dans ladite branche afin que votre branche ne diverge pas plus que nécessaire. Dans le répertoire de plus haut niveau de votre branche, veuillez en premier taper `‘svn info‘` afin d'obtenir la version de révision de votre branche ce qui doit afficher quelque chose similaire à :

```
timlinux@timlinux-desktop:~/dev/cpp/qgis_raster_transparency_branch$ svn info
Caminho: .
URL: https://svn.qgis.org/repos/qgis/branches/raster_transparency_branch
Raiz do Repositorio: https://svn.qgis.org/repos/qgis
UUID do repositorio: c8812cc2-4d05-0410-92ff-de0c093fc19c
Revisao: 6546
Tipo de No: diretorio
Agendado: normal
Autor da Ultima Mudanca: timlinux
Revisao da Ultima Mudanca: 6495
Data da Ultima Mudanca: 2007-02-02 09:29:47 -0200 (Sex, 02 Fev 2007)
Propriedades da Ultima Mudanca: 2007-01-09 11:32:55 -0200 (Ter, 09 Jan 2007)
```

Le second indice de révisions indique la version d'origine de votre branche tandis que le premier vous donne la version en cours. Vous pouvez lancer une simulation de fusion comme suit :

```
svn merge --dry-run -r 6495:6546 https://svn.qgis.org/repos/qgis/trunk/qgis
```

Après être satisfait des changements à effectuer, vous pouvez lancer la véritable fusion comme suit :

```
svn merge -r 6495:6546 https://svn.qgis.org/repos/qgis/trunk/qgis
svn commit -m "Merged upstream changes from trunk to my branch"
```

K.6 Soumettre des correctifs

Il existe quelques aides pouvant vous aider pour avoir aisément vos correctifs dans QGIS, et nous aider à gérer facilement les correctifs envoyés.

K.6.1 Nommage des fichiers du correctif

Si le correctif permet de résoudre un bug spécifique, veuillez nommer le fichier en intégrant le numéro du bug, e.g. **bug777fix.diff**, et attacher le fichier au rapport original du bug dans le trac (https ://trac.osgeo.org/qgis/).

Si le bug est une nouvelle fonctionnalité ou une amélioration, il est classiquement de bon goût de créer en premier lieu un ticket dans le trac (https ://trac.osgeo.org/qgis/) et ensuite d'attacher votre correctif.

K.6.2 Créer votre correctif dans le dossier de plus haut niveau des sources de QGIS

Ainsi, il est plus facile pour nous d'appliquer les correctifs puisque nous n'avons pas besoin de nous placer dans des répertoires spécifiques dans l'arbre des sources pour appliquer les correctifs. De plus, quand je reçois des correctifs, j'ai l'habitude de les évaluer en utilisant kompare ; placer le correctif dans le dossier de plus haut niveau facilite grandement cette exploitation. Vous trouverez ci-dessous un exemple vous permettant d'inclure les fichiers du correctif à partir du dossier de plus haut niveau :

```
cd qgis
svn diff src/ui/somefile.ui src/app/somefile2.cpp > bug872fix.diff
```

K.6.3 Inclure une version non controllée des fichiers dans votre correctif

Si vos corrections contiennent de nouveaux fichiers encore non existants dans le répertoire, vous devez indiquer dans le svn que ces fichiers doivent être ajoutés avant de générer votre correctif, p. ex.

```
cd qgis
svn add src/lib/somenewfile.cpp
svn diff > bug7887fix.diff
```

K.6.4 Obtenir une notification de votre correctif

Les développeurs de QGIS sont très occupés. Nous scannons les correctifs des bugs rapportés arrivant, mais il peut nous arriver d'en oublier. N'en soyez pas offensé ou alarmé. Tentez d'identifier des développeurs pour vous aider, en utilisant le ["Project Organigram"], et contacter les pour leur demander s'ils peuvent regarder votre correctif. Si vous n'obtenez aucune réponse, vous pouvez faire remonter votre requête aux membres du Project Steering Committee dont les contacts sont disponibles sur le ["Project Organigram"].

K.6.5 Diligence des droits

QGIS est sous licence GPL. Vous devez faire tous les efforts nécessaires pour vous assurer que vous soumettez uniquement des correctifs qui ne peuvent être en conflit avec les droits à la propriété intellectuelle. Ainsi, veuillez ne pas soumettre du code pour lequel la disponibilité sous GPL ne vous satisferez pas.

K.7 Obtenir les droits en écriture sur le SVN

Les droits en écriture sur les sources de QGIS s'obtiennent sur invitation. Typiquement, quand une personne soumet plusieurs (il n'y a pas de nombre fixé) correctifs substantiels qui démontrent une base de compétence et de compréhension du C++ et des conventions de code dans QGIS, un membre du PSC ou d'autres développeurs peuvent proposer cette personne au PSC pour l'obtention des droits en écriture. La personne présentant le candidat doit fournir une courte présentation motivant le choix de la personne. Dans certains cas, nous donnons les droits d'écriture à des personnes ne développant pas en C++ à l'instar des traducteurs et des documentalistes. Dans ces cas, la personne doit quand même démontrer sa capacité à soumettre des correctifs et doit dans l'idéal avoir soumis des correctifs substantiels démontrant sa compréhension des modifications du code de base sans perturbations.

K.7.1 Procédure quand vous avez les droits en écriture

Vérifiez les sources :

```
svn co https://svn.qgis.org/repos/qgis/trunk/qgis qgis
```

Compilez les sources (voir le document INSTALL pour les instructions détaillées)

```
cd qgis
```

```
mkdir build
ccmake ..    (set your preferred options)
make
make install  (maybe you need to do with sudo / root perms)
```

Faites vos éditions

```
cd ..
```

Faites vos corrections dans les sources. Il faut toujours vérifier que tout compile avant de faire quoi que ce soit. Tâchez de vérifier les possibles perturbations que vos modifications peuvent causer aux personnes compilant sur d'autres plateformes ou avec des versions plus anciennes ou récentes des bibliothèques.

Ajoutez les fichiers (si vous avez ajouté de nouveaux fichiers). La commande pour le statut du svn peut être utilisée pour voir rapidement si vous avez ajouté de nouveaux fichiers.

```
svn status src/pluguns/grass/modules
```

Les fichiers listés avec ? au début ne sont pas dans le SVN et il possible qu'il soit nécessaire qu'ils soient ajoutés par vous :

```
svn add src/pluguns/grass/modules/foo.xml
```

Réalisez vos changements

```
svn commit src/pluguns/grass/modules/foo.xml
```

Votre éditeur (celui défini dans la variable d'environnement $EDITOR) apparaitra et vous devrez faire un commentaire en haut du fichier (au-dessus de la zone disant 'dont change this'). Donnez un commentaire descriptif de votre correction et faire aussi des commentaires courts au travers des fichiers non reliés. Nous préférons que vous groupiez vos commentaires relatifs à vos modifications dans un unique commentaire.

Sauvez et fermez votre éditeur. La première fois que vous le faites, vous devez être invité à saisir votre nom d'utilisateur et votre mot de passe. Utilisez les mêmes que ceux de votre compte trac.

L Les Tests unitaires

Depuis novembre 2007, nous exigeons que toutes les nouvelles fonctionnalités à intégrer à la branche principale de développement soient accompagnées d'un test unitaire. Cette exigence était initialement limitée à qgis_core, nous l'étendrons à d'autres parties du code dès que les gens seront plus familiers avec les procédures décrites dans le reste de cette section.

L.1 L'environnement de test de QGIS - un aperçu

Les tests unitaires sont véhiculés par une combinaison de QTestLib (la bibliothèque de test de Qt) et de CTest (un environnement de compilation et d'exécution de tests faisant partie de CMake). Prenons un aperçu de ce processus avant de rentrer dans les détails :

– **Ici se trouve le code à tester**, p. ex. une classe ou une fonction. Les partisans de l'Extreme Programming suggèrent que le code ne devrait pas être écrit avant d'avoir conçu les tests, ainsi vous pouvez valider votre code au fur et à mesure de son implémentation. Dans la pratique, vous aurez probablement besoin d'écrire des tests pour du code pré-existant dans QGIS du fait que nous avons entrepris cet effort de test bien après avoir réalisé de nombreuses applications logiques.
– **Vous créez un test unitaire.** Cela se produit dans <QGIS Source Dir>/tests/src/core pour le cas d'une bibliothèque principale. Le test est un client basique qui créé une instance d'une classe et appelle quelques méthodes de cette même classe. Il vérifie ce que retourne chaque méthode pour s'assurer que cela correspond bien au résultat attendu. Si l'un de ces appels échoue alors le test échoue également.
– **Vous incluez une macro QtTestLib dans votre classe de test.** Cette macro est traitée par le compilateur de méta-objet de Qt (moc) et étend votre classe de test vers une application exécutable.
– **Vous ajoutez une section à CMakeLists.txt** dans votre répertoire de tests qui construira votre test.
– **Vous vous assurez d'avoir activé ENABLE_TESTING dans ccmake / cmakesetup.** Cela permet de garantir que vos tests seront bien compilés lorsque vous taperez make.
– **Vous pouvez ajoutez les données du test dans <QGIS Source Dir>/tests/testdata** si votre test nécessite des données (p. ex. nécessité de charger un shapefile). Ces données doivent être aussi légères que possible et quand c'est possible, réutiliser les données déjà présentes. Vos tests ne doivent jamais modifier les données en place mais plutôt une copie temporaire.
– **Vous compilez vos sources et installez.** Faites-le avec une procédure normale de make && (sudo) make install.
– **Vous exécutez vos tests.** C'est normalement effectué par **make test** après l'étape make install, je vous expliquerais d'autres approches qui offrent plus de contrôles sur l'exécution.

Avec cet aperçu en tête, je vais vous expliquer plus en détail avec une configuration déjà préparée pour CMake et d'autres parties des sources de manière à ce que vous n'ayez qu'à faire les choses

les plus simples - écrire des tests unitaires !

L.2 Créer un test unitaire

Créer un test unitaire est simple - vous allez généralement le faire en créant un seul fichier .cpp (aucun .h n'est utilisé) et implémenter vos méthodes de test en tant que méthodes publiques qui ne retourneront rien. Pour illustrer ce point, je vais utiliser une classe de test pour QgsRasterLayer. Par convention, nous allons appeler ce test avec le même nom que la classe qui va être testé, mais en rajoutant le préfixe 'test'. Notre exemple sera donc un fichier testqgsrasterlayer.cpp et une classe TestQgsRasterLayer. Il faut d'abord rajouter un en-tête précisant la licence et l'auteur :

```
/***************************************************************************
     testqgsvectorfilewriter.cpp
     --------------------------------------
    Date                 : Frida  Nov 23  2007
    Copyright            : (C) 2007 by Tim Sutton
    Email                : tim@linfiniti.com
 ***************************************************************************
 *                                                                         *
 *   This program is free software; you can redistribute it and/or modify  *
 *   it under the terms of the GNU General Public License as published by  *
 *   the Free Software Foundation; either version 2 of the License, or     *
 *   (at your option) any later version.                                   *
 *                                                                         *
 ***************************************************************************/
```

Ensuite nous allons débuter par les inclusions (includes) nécessaires aux tests que nous voulons faire.Next we use start our includes needed for the tests we plan to run. Il y a une spéciale que tous les tests vont avoir :

```
#include <QtTest>
```

A par ça, vous pouvez écrire vos classes comme d'habitude, en utilisant n'importe quel en-tête (header) que vous voulez :

```
//Qt includes...
#include <QObject>
#include <QString>
#include <QObject>
#include <QApplication>
#include <QFileInfo>
#include <QDir>

//qgis includes...
#include <qgsrasterlayer.h>
#include <qgsrasterbandstats.h>
#include <qgsapplication.h>
```

Puisque nous combinons la déclaration et l'implémentation de classe dans un unique fichier, la déclaration vient ensuite. On débute avec la documentation doxygen. Chaque test doit être proprement documenté, nous utilisons la directive Doxygen **ingroup** pour que l'ensemble des tests unitaires

(UnitTests) apparaisse comme un module dans la documentation Doxygen générée. C'est suivi d'une courte description du test :

```
/** \ingroup UnitTests
 * This is a unit test for the QgsRasterLayer class.
 */
```

La classe **doit** hériter de QObject et inclure la macro Q_OBJECT.

```
class TestQgsRasterLayer: public QObject
{
  Q_OBJECT;
```

Toutes nos méthodes sont conçues comme des **slots privés**. L'environnement QtTest appellera séquentiellement chacun des slots privés de la classe de test. Il y a quatre méthodes 'spéciales' qui si utilisées seront appelées au début de chaque test unitaire (**initTestCase**),et à leur fin (**cleanupTestCase**). Avant que chaque méthode soit appelée, la méthode **init()** sera appelée et après ce sera la méthode **cleanup()**. Ces méthodes sont pratiques, car elles vous permettent d'allouer et de nettoyer les ressources avant d'exécuter chaque test.

```
private slots:
    // will be called before the first testfunction is executed.
    void initTestCase();
    // will be called after the last testfunction was executed.
    void cleanupTestCase(){};
    // will be called before each testfunction is executed.
    void init(){};
    // will be called after every testfunction.
    void cleanup();
```

Ensuite viennent vos méthodes de test, toutes doivent être **sans paramètres** et ne **rien retourner**. Elles sont appelées dans l'ordre de déclaration. Voici deux exemples, dans le premier cas je veux tester si les diverses parties de la classe fonctionnent, je peux utilise une approche de **test fonctionnel**. Une fois encore, certains sont partisans d'écrire ces tests **avant** de créer la classe puis de concevoir la classe en exécutant de façon itérative votre test unitaire. De plus en plus de fonctions de test devraient réussir au long de l'écriture de la classe et quand le test entier réussit votre nouvelle classe est complète avec un test réutilisable pour la valider. Habituellement, votre test unitaire ne devrait couvrir seulement l'API **publique**, vous ne devriez pas avoir à écrire de tests pour un accessoire ou un mutator. Si l'un d'eux ne fonctionne pas comme prévu, vous devrez créer un test de **régression** pour le vérifier.

```
//
// Functional Testing
//

/** Check if a raster is valid. */
void isValid();

// more functional tests here ...
```

Maintenant nous créons nos **tests de régression**, ils doivent répliquer les conditions d'un bug particulier . Par exemple, j'ai récemment reçu un rapport de bug par email sur le fait que le compte de cellules d'un raster était décalé de 1, faussant toutes les statistiques pour les bandes de rasters/ J'ai ouvert un rapport de bug (ticket #832) et créé un test de régression qui répliquait le bug en utilisant un petit raster de 10x10. Puis j'ai exécuté le test à plusieurs reprises, en vérifiant qu'il échouait bien (le compte des cellules était de 99 au lieu de 100). J'ai corrigé ce bug et relancé le test avec succès. J'ai joint ce test à la correction pour s'assurer que si jamais ce bug devait se présenter de nouveau dans le code source nous pourrions l'identifier immédiatement. Encore mieux, en exécutant ce test avant chaque changement nous nous assurons qu'ils n'auront pas d'effets indésirables - comme briser une fonctionnalité existante.

Il y a un autre bénéfice à ces tests de régression - ils peuvent vous faire gagner du temps. Si vous fixez un bug nécessitant un changement dans les sources puis lancez une série de vérifications complexes dans l'application pour vous assurer de sa résolution, il vous sera vite évident que l'écriture de ce test de régression **avant** de fixer le bug vous permet d'automatiser le test de manière efficace.

Pour créer votre test de régression, vous pouvez suivre la convention de nommage regression<TicketID> pour vos fonctions. Si aucun ticket de suivi n'existe pour cette régression, vous devrez en créer un. Cette approche permet à une personne constatant l'échec d'un test de régression d'obtenir plus d'informations.

```
//
// Regression Testing
//

/** This is our second test case...to check if a raster
 reports its dimensions properly. It is a regression test
 for ticket #832 which was fixed with change r7650.
 */
void regression832();

// more regression tests go here ...
```

Finalement dans notre déclaration de classe de test, vous allez déclarer de manière privée toutes les données chiffrées et les méthodes que votre test pourrait nécessiter. Dans notre cas, je vais déclarer QgsRasterLayer * qui peut être utilisé par chacune des méthodes de test. La couche raster sera créer par la fonction initTestCase() puis détruire par cleanupTestCase(). En déclarant les méthodes assistantes (qui peuvent être utilisé par différentes fonctions de test) de manière privée, vous pouvez vous assurer qu'elles ne seront pas automatiquement lancées par l'exécutable QTest créé quand nous compilons notre test.

```
  private:
    // Here we have any data structures that may need to
    // be used in many test cases.
    QgsRasterLayer * mpLayer;
};
```

Cela clôt notre déclaration de classe. Cette implémentation est simplement placée plus bas dans le même fichier. D'abord nos fonctions init et cleanup :

```
void TestQgsRasterLayer::initTestCase()
{
  // init QGIS's paths - true means that all path will be inited from prefix
  QString qgisPath = QCoreApplication::applicationDirPath ();
  QgsApplication::setPrefixPath(qgisPath, TRUE);
#ifdef Q_OS_LINUX
  QgsApplication::setPkgDataPath(qgisPath + "/../share/qgis");
#endif
  //create some objects that will be used in all tests...

  std::cout << "Prefix  PATH: " << QgsApplication::prefixPath().toLocal8Bit().data() \
  << std::endl;
  std::cout << "Plugin  PATH: " << QgsApplication::pluginPath().toLocal8Bit().data() \
  << std::endl;
  std::cout << "PkgData PATH: " << QgsApplication::pkgDataPath().toLocal8Bit().data() \
  << std::endl;
  std::cout << "User DB PATH: " << QgsApplication::qgisUserDbFilePath().toLocal8Bit() \
  .data() << std::endl;

  //create a raster layer that will be used in all tests...
  QString myFileName (TEST_DATA_DIR); //defined in CmakeLists.txt
  myFileName = myFileName + QDir::separator() + "tenbytenraster.asc";
  QFileInfo myRasterFileInfo ( myFileName );
```

```
  mpLayer = new QgsRasterLayer ( myRasterFileInfo.filePath(),
            myRasterFileInfo.completeBaseName() );
}

void TestQgsRasterLayer::cleanupTestCase()
{
  delete mpLayer;
}
```

La fonction init ci-dessus démontre quelques points intéressants :

1. J'ai eu besoin de mettre manuellement le chemin des données de QGIS pour que les ressources telles que srs.db soient correctement détectées. 2. Deuxièmement, c'est un test recourant à des données donc nous devons fournir une façon de localiser le fichier tenbytenraster.asc. Cela est fait en définissant via le compilateur **TEST_DATA_PATH**. Cette définition est créé dans le fichier de configuration CMakeLists.txt dans <QGIS Source Root>/tests/CMakeLists.txt et est disponible pour tous les tests unitaires de QGIS. Si vous avez besoin d'autres fichiers, placez-les dans <QGIS Source Root>/tests/testdata. Efforcez-vous de réduire leur poids et surtout n'écrivez que sur des copies.

Qt fourni également des mécanismes utiles pour les tests utilisant des données, si cela vous intéresse consultez la documentation.

Maintenant, regardons notre test fonctionnel. Le test isValid() vérifie simplement si la couche raster est correctement chargée par initTestCase. QVERIFY est une macro Qt que vous pouvez utiliser pour évaluer une condition de test. Il y a d'autres macros que vous pouvez utiliser :

```
QCOMPARE ( actual, expected )
QEXPECT_FAIL ( dataIndex, comment, mode )
QFAIL ( message )
QFETCH ( type, name )
QSKIP ( description, mode )
QTEST ( actual, testElement )
QTEST_APPLESS_MAIN ( TestClass )
QTEST_MAIN ( TestClass )
QTEST_NOOP_MAIN ()
QVERIFY2 ( condition, message )
QVERIFY ( condition )
QWARN ( message )
```

Certaines de ces macros ne sont utiles que si vous utilisez l'environnement Qt pour les tests recourant aux données (voir la documentation de Qt).

```
void TestQgsRasterLayer::isValid()
{
  QVERIFY ( mpLayer->isValid() );
}
```

Normalement vos tests fonctionnels devraient couvrir l'ensemble des fonctionnalités de vos classes. Ayant vu cela nous pouvons passer à notre exemple de test de régression.

Le problème #832 est un compte de cellules mal effectué, l'écriture de notre test va simplement consisté à utiliser QVERIFY pour regarder si la valeur calculer est bien la valeur qui était prévue :

```
void TestQgsRasterLayer::regression832()
{
   QVERIFY ( mpLayer->getRasterXDim() == 10 );
   QVERIFY ( mpLayer->getRasterYDim() == 10 );
   // regression check for ticket #832
   // note getRasterBandStats call is base 1
   QVERIFY ( mpLayer->getRasterBandStats(1).elementCountInt == 100 );
}
```

Avec toutes les fonctions du test crées, il reste encore une chose à ajouter pour finaliser notre classe :

```
QTEST_MAIN(TestQgsRasterLayer)
#include "moc_testqgsrasterlayer.cxx"
```

Le but de ces deux lignes est de signaler au moc de Qt que c'est un QTest (ça génèrera une méthode principale qui appellera chaque fonction du test). La dernière ligne est une inclusion pour les sources générées du MOC. Vous devez remplacer 'testqgsrasterlayer' par le nom de votre classe en caractère minuscule.

L.3 Ajouter votre test unitaire à CMakeLists.txt

Ajouter votre test unitaire au système de construction consiste juste à éditer le fichier CMakeLists.txt, copier un des blocs de test existants, puis de remplacer par votre nom de classe. Par exemple :

```
#
# QgsRasterLayer test
#
SET(qgis_rasterlayertest_SRCS testqgsrasterlayer.cpp)
```

```
SET(qgis_rasterlayertest_MOC_CPPS testqgsrasterlayer.cpp)
QT4_WRAP_CPP(qgis_rasterlayertest_MOC_SRCS ${qgis_rasterlayertest_MOC_CPPS})
ADD_CUSTOM_TARGET(qgis_rasterlayertestmoc ALL DEPENDS ${qgis_rasterlayertest_MOC_SRCS})
ADD_EXECUTABLE(qgis_rasterlayertest ${qgis_rasterlayertest_SRCS})
ADD_DEPENDENCIES(qgis_rasterlayertest qgis_rasterlayertestmoc)
TARGET_LINK_LIBRARIES(qgis_rasterlayertest ${QT_LIBRARIES} qgis_core)
INSTALL(TARGETS qgis_rasterlayertest RUNTIME DESTINATION ${QGIS_BIN_DIR})
ADD_TEST(qgis_rasterlayertest ${QGIS_BIN_DIR}/qgis_rasterlayertest)
```

Je vais parcourir brièvement ces lignes pour expliquer leur but, mais si cela ne vous intéresse pas, copiez le bloc et faites une recherche et puis un remplacement

```
:'<,'>s/rasterlayer/mynewtest/g
```

Regardons en détail ces lignes. premièrement, nous définissons la liste des sources pour notre test et puisque nous n'avons qu'un fichier source (en ayant suivi la méthodologie décrite précédemment où la déclaration et la définition d'une classe sont dan s le même fichier), c'est une ligne simple :

```
SET(qgis_rasterlayertest_SRCS testqgsrasterlayer.cpp)
```

Notre classe de test requiert d'être lancé via le compilateur méta-objet de Qt (MOC), nous devons fournir un couple de ligne pour le permettre :

```
SET(qgis_rasterlayertest_MOC_CPPS testqgsrasterlayer.cpp)
QT4_WRAP_CPP(qgis_rasterlayertest_MOC_SRCS ${qgis_rasterlayertest_MOC_CPPS})
ADD_CUSTOM_TARGET(qgis_rasterlayertestmoc ALL DEPENDS ${qgis_rasterlayertest_MOC_SRCS})
```

Ensuite, nous devons signifier à CMake qu'il doit produire un exécutable à partir de la classe de test. Souvenez-vous que dans la section précédente j'ai inclus à la dernière ligne le résultat du MOC directement dans notre classe, donc cela donnera (entre autres) une méthode principale qui permettra à la classe d'être compilée en tant qu'exécutable :

```
ADD_EXECUTABLE(qgis_rasterlayertest ${qgis_rasterlayertest_SRCS})
ADD_DEPENDENCIES(qgis_rasterlayertest qgis_rasterlayertestmoc)
```

A présent, nous allons spécifier les dépendances de bibliothèques. Pour l'instant les classes ont été réalisées avec une dépendance sur toutes les QT_LIBRARIES, mais je vais réduire ce montant aux bibliothèques réellement utilisées par les classes. Bien entendu vous devez lier les bibliothèques qgis requises par votre test unitaire.

```
TARGET_LINK_LIBRARIES(qgis_rasterlayertest ${QT_LIBRARIES} qgis_core)
```

Il faut maintenant dire à Cmake de se placer dans le même répertoire que les binaires de QGIS. C'est un point que je prévois de supprimer dans le futur pour que les tests puissent être lancés depuis l'arbre des sources.

```
INSTALL(TARGETS qgis_rasterlayertest RUNTIME DESTINATION ${QGIS_BIN_DIR})
```

Voici le moment magique - nous enregistrons la classe avec ctest. Si vous vous rappelez l'aperçu du début, nous utilisons conjointement QtTest et CTest. Pour récapituler, **QtTest** ajoute une méthode principale à notre test et gère les appels des méthodes internes à la classe. Il fournit aussi des macros comme QVERIFY pour vérifier les échecs à des conditions pré-établies. Le résultat est un exécutable que vous pouvez lancer depuis la console. Il est important quand vous avez une série de tests que vous voulez effectuer, ou mieux encore, que vous voulez intégrer au processus de compilation, d'utiliser **CTest**. La ligne suivante enregistre le test unitaire avec CMake / CTest.

```
ADD_TEST(qgis_rasterlayertest ${QGIS_BIN_DIR}/qgis_rasterlayertest)
```

La dernière chose que j'ajouterai est que si votre test requiert des composants optionnels (p. ex. support de PostgreSQL, GRASS, etc.), vous devrez faire attention à circoncire votre bloc de test dans un IF () au sein du fichier CMakeLists.txt.

L.4 Compiler votre test unitaire

Pour construire votre test unitaire, vous devez seulement vous assurer que ENABLE_TESTS=true dans la configuration de cmake ; vous avez deux possibilités pour ce faire : 1. Lancer ccmake .. (cmakesetup .. sous windows) et mettre ENABLE_TESTS sur ON. 2. Ajouter une ligne de commande à cmake e.g. cmake -DENABLE_TESTS=true .. À part ça, compilez QGIS comme d'habitude et les tests devraient être compilés également.

L.5 Exécuter vos tests

La façon la plus simple est de les lancer au court du processus de construction normal :

```
make && make install && make test
```

La commande make test va invoquer CTest pour exécuter chaque test qui a été enregistré avec la directive CMake ADD_TEST. Le résultat typique ressemble à ça :

```
Running tests...
Start processing tests
Test project /Users/tim/dev/cpp/qgis/build
1/  3 Testing qgis_applicationtest         ***Exception: Other
2/  3 Testing qgis_filewritertest          *** Passed
3/  3 Testing qgis_rasterlayertest         *** Passed

0% tests passed, 3 tests failed out of 3

  The following tests FAILED:
  1 - qgis_applicationtest (OTHER_FAULT)
  Errors while running CTest
  make: *** [test] Error 8
```

Si un test échoue, vous pouvez utiliser la commande ctest pour examiner de plus prêt la raison de l'échec. Utilisez l'option -R pour spécifier un regex pour les tests que vous voulez exécuter et -V pour afficher une sortie verbeuse :

```
[build] ctest -R appl -V
Start processing tests
Test project /Users/tim/dev/cpp/qgis/build
Constructing a list of tests
Done constructing a list of tests
Changing directory into /Users/tim/dev/cpp/qgis/build/tests/src/core
1/  3 Testing qgis_applicationtest
Test command: /Users/tim/dev/cpp/qgis/build/tests/src/core/qgis_applicationtest
********* Start testing of TestQgsApplication *********
  Config: Using QTest library 4.3.0, Qt 4.3.0
PASS   : TestQgsApplication::initTestCase()
  Prefix  PATH: /Users/tim/dev/cpp/qgis/build/tests/src/core/../
  Plugin  PATH: /Users/tim/dev/cpp/qgis/build/tests/src/core/..//lib/qgis
  PkgData PATH: /Users/tim/dev/cpp/qgis/build/tests/src/core/..//share/qgis
  User DB PATH: /Users/tim/.qgis/qgis.db
PASS   : TestQgsApplication::getPaths()
  Prefix  PATH: /Users/tim/dev/cpp/qgis/build/tests/src/core/../
  Plugin  PATH: /Users/tim/dev/cpp/qgis/build/tests/src/core/..//lib/qgis
  PkgData PATH: /Users/tim/dev/cpp/qgis/build/tests/src/core/..//share/qgis
  User DB PATH: /Users/tim/.qgis/qgis.db
  QDEBUG : TestQgsApplication::checkTheme() Checking if a theme icon exists:
  QDEBUG : TestQgsApplication::checkTheme()
  /Users/tim/dev/cpp/qgis/build/tests/src/core/.. \
```

```
  //share/qgis/themes/default//mIconProjectionDisabled.png
  FAIL!  : TestQgsApplication::checkTheme() '!myPixmap.isNull()' returned FALSE. ()
  Loc: [/Users/tim/dev/cpp/qgis/tests/src/core/testqgsapplication.cpp(59)]
PASS   : TestQgsApplication::cleanupTestCase()
  Totals: 3 passed, 1 failed, 0 skipped
  ********* Finished testing of TestQgsApplication *********
  -- Process completed
  ***Failed

  0% tests passed, 1 tests failed out of 1

  The following tests FAILED:
1 - qgis_applicationtest (Failed)
  Errors while running CTest
```

Ceci conclut la section sur les tests unitaires dans QGIS. Nous espérons que vous allez vous habituer à écrire des tests pour éprouver chaque fonctionnalité et repérer les régressions. Certains aspects du système de test sont encore en chantier (notamment la partie sur CMakeLists.txt) pour améliorer l'environnement de test, le document rendra compte de ces mies à jour dans la procédure.

M HIG (Guide de l'Interface Humaine)

Pour avoir une interface graphique dont les divers éléments aient une consistance et que les utilisateurs puissent utiliser de manière instinctive, il est important de respecter les principes énoncés ci-dessous dans le design et la conception de l'interface.

1. Grouper les éléments liés en utilisant des boîtes de groupe : Essayez d'identifier les éléments qui peuvent être rassemblés puis utilisez des boîtes munies d'une étiquette visant à déterminer le rôle de ce groupe. Il faut éviter de créer des boîtes avec un seul élément à l'intérieur.
2. Mettre en majuscule la première lettre uniquement dans les étiquettes : Les étiquettes (et celles des boîtes) doivent être rédigées avec une phrase dont la première lettre est en majuscule, tous les mots qui suivent doivent être en minuscule.
3. Ne pas terminer les étiquettes avec une colonne : L'ajout d'une colonne provoque une nuisance visuelle et ne donne aucune signification supplémentaire. Une exception à cette règle est si vous avez deux étiquettes côte à côte p. ex. : Etiquette1 Extension Etiquette2 [/chemin/vers/extensions]
4. Placer les actions dangereuses à l'écart de celles inoffensives : Si vous avez une action pour 'effacer', 'annuler', etc., essayez de conserver un espace entre ces actions et celles moins à risque pour éviter que l'utilisateur clique dessus par mégarde.

5. Toujours utiliser une QButtonBox pour les boutons 'OK', 'Annuler', etc. : Cela permet d'avoir un ordre et une apparence des boutons qui soient cohérents avec la plateforme / langue / environnement de bureau de l'utilisateur.

N GNU General Public License

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs ; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps : (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law : that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License ; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty ; keep intact all the notices that refer to this License and to the absence of any warranty ; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions :

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception : if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you ; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under

the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following :

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange ; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange ; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from

distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims ; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system ; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation ; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPE-

CIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

N.1 Quantum GIS Qt exception for GPL

In addition, as a special exception, the QGIS Development Team gives permission to link the code of this program with the Qt library, including but not limited to the following versions (both free and commercial) : Qt/Non-commerical Windows, Qt/Windows, Qt/X11, Qt/Mac, and Qt/Embedded (or with modified versions of Qt that use the same license as Qt), and distribute linked combinations including the two. You must obey the GNU General Public License in all respects for all of the code used other than Qt. If you modify this file, you may extend this exception to your version of the file, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

Littérature

Références Web